

Ε.Π. Εκπαίδευση και Δια Βίου Μάθηση, ΕΣΠΑ (2007 – 2013)

ΕΠΙΜΟΡΦΩΣΗ ΕΚΠΑΙΔΕΥΤΙΚΩΝ ΓΙΑ ΤΗΝ ΑΞΙΟΠΟΙΗΣΗ ΚΑΙ ΕΦΑΡΜΟΓΗ ΤΩΝ ΤΠΕ ΣΤΗ ΔΙΔΑΚΤΙΚΗ ΠΡΑΞΗ

Επιμορφωτικό υλικό
για την επιμόρφωση των εκπαιδευτικών στα
Κέντρα Στήριξης Επιμόρφωσης

ΤΕΥΧΟΣ 6Α: ΕΙΔΙΚΟ ΜΕΡΟΣ ΚΛΑΔΩΝ ΠΕ19/20 ΘΕΩΡΙΑ ΔΙΔΑΚΤΙΚΗΣ ΤΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

Α' έκδοση



Διεύθυνση Επιμόρφωσης & Πιστοποίησης

Πάτρα, Μάρτιος 2013



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Περιεχόμενα

Προοίμιο	5
Ενότητα ΘΕΩΡΙΑ ΔΙΔΑΚΤΙΚΗΣ ΤΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ 1	7
Εισαγωγή στη Διδακτική της Πληροφορικής – διδασκαλία του προγραμματισμού	7
1. ΣΥΝΟΠΤΙΚΗ ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΕΠΙΜΟΡΦΩΤΙΚΟΥ ΣΕΝΑΡΙΟΥ	7
2. ΕΚΤΙΜΩΜΕΝΗ ΔΙΑΡΚΕΙΑ ΕΠΙΜΟΡΦΩΤΙΚΟΥ ΣΕΝΑΡΙΟΥ	7
3. ΣΚΟΠΟΙ ΚΑΙ ΣΤΟΧΟΙ ΤΟΥ ΕΠΙΜΟΡΦΩΤΙΚΟΥ ΣΕΝΑΡΙΟΥ	7
4. ΘΕΩΡΗΤΙΚΕΣ ΕΝΝΟΙΕΣ ΑΠΟ ΤΗ ΔΙΔΑΚΤΙΚΗ ΤΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ.....	8
Η ανάγκη για ένα θεωρητικό πλαίσιο	8
Οι αντιλήψεις των σπουδαστών	10
Απλοϊκές αντιλήψεις: το παράδειγμα του ανθρωπομορφισμού	11
Αντιλήψεις σχετιζόμενες με τη χρήση επαναληπτικών διαδικασιών	12
Η αναδρομή	13
Οι ακαδημαϊκοί και οι επαγγελματίες προγραμματιστές: η σχέση με τα «λάθη»	14
Η διδασκόμενη γνώση κι ο μετασχηματισμός της.....	19
Το διδακτικό σύμβολο	24
Η βασική υπόθεση της Διδακτικής	26
Η Διδακτική ως πειραματική επιστήμη.....	27
Σύνοψη	29
Οι σπουδαστές.....	30
Μερικές πρόσθετες παρατηρήσεις	31
Οι παλαιότερες εργασίες.....	33
Πιο σύγχρονες εργασίες.....	35
5. ΟΙ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟΙ ΜΙΚΡΟΚΟΣΜΟΙ: ΜΙΑ ΓΕΝΙΚΗ ΘΕΩΡΗΣΗ.....	36
Μικρόκοσμοι – Μικρογλώσσες Προγραμματισμού	39
Προσομοιώσεις φυσικών αντικειμένων που λειτουργούν ως μικρόκοσμοι	41
Παιχνίδια ως μικρόκοσμοι	41
Η «οικογένεια» περιβαλλόντων Logo (Logo-like environments)	42
6. ΠΑΡΑΔΕΙΓΜΑΤΑ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΩΝ ΜΙΚΡΟΚΟΣΜΩΝ ΚΑΙ ΕΝΑΛΛΑΚΤΙΚΩΝ ΠΕΡΙΒΑΛΛΟΝΤΩΝ.....	43
Περιβάλλοντα έκφρασης αλγορίθμων χωρίς Η.Υ.	44
Περιβάλλοντα δημιουργίας παιχνιδιών	46
Άλλα εναλλακτικά περιβάλλοντα	47
7. ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ ΚΑΙ ΕΡΓΑΣΙΕΣ	48
8. ΧΡΗΣΗ ΕΞΩΤΕΡΙΚΩΝ ΠΗΓΩΝ	56

ΒΙΒΛΙΟΓΡΑΦΙΑ (ενδεικτική)	56
ΔΙΚΤΥΟΓΡΑΦΙΑ	60
ΙΣΤΟΧΩΡΟΙ.....	60
9. ΠΑΡΑΡΤΗΜΑ 1: Ο ΧΑΡΤΙΝΟΣ ΥΠΟΛΟΓΙΣΤΗΣ	60
10. ΠΑΡΑΡΤΗΜΑ 2: Η ΜΗΧΑΝΗ ΤΟΥ POST.....	76
Εισαγωγή	76
Η μηχανή του Post	78
11. ΠΑΡΑΡΤΗΜΑ 3: MORTRAN	81
Προτάσεις για τη δημιουργία διδακτικών σεναρίων με το MORTRAN.....	83
Ενότητα ΘΕΩΡΙΑ ΔΙΔΑΚΤΙΚΗΣ ΤΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ 2	86
Διδακτικά σενάρια – τι είναι και πώς δημιουργούνται	86
ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΕΠΙΜΟΡΦΩΤΙΚΟΥ ΣΕΝΑΡΙΟΥ.....	86
ΕΚΤΙΜΩΜΕΝΗ ΔΙΑΡΚΕΙΑ ΕΠΙΜΟΡΦΩΤΙΚΟΥ ΣΕΝΑΡΙΟΥ.....	86
ΣΚΟΠΟΙ ΚΑΙ ΣΤΟΧΟΙ ΤΟΥ ΕΠΙΜΟΡΦΩΤΙΚΟΥ ΣΕΝΑΡΙΟΥ.....	86
ΘΕΩΡΗΤΙΚΕΣ ΕΝΝΟΙΕΣ ΑΠΟ ΤΗ ΔΙΔΑΚΤΙΚΗ ΤΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ	87
Το μοντέλο της Γνώσης Τεχνολογικής Παιδαγωγικής Περιεχομένου (ΓΤΠΠ – TRACK)	87
Το εκπαιδευτικό σενάριο με ΤΠΕ	89
Φάσεις ανάπτυξης εκπαιδευτικού σεναρίου	102
ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ ΚΑΙ ΕΡΓΑΣΙΕΣ	117
ΕΡΩΤΗΣΕΙΣ ΚΛΕΙΣΤΟΥ ΤΥΠΟΥ	118
ΧΡΗΣΗ ΕΞΩΤΕΡΙΚΩΝ ΠΗΓΩΝ	119
ΒΙΒΛΙΟΓΡΑΦΙΑ	119
ΔΙΚΤΥΟΓΡΑΦΙΑ	120
ΙΣΤΟΧΩΡΟΙ	120
Ενότητα ΘΕΩΡΙΑ ΔΙΔΑΚΤΙΚΗΣ ΤΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ 3	121
Διδακτική αντικειμενοστρεφούς προγραμματισμού	121
1. ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΕΠΙΜΟΡΦΩΤΙΚΟΥ ΣΕΝΑΡΙΟΥ	121
2. ΕΚΤΙΜΩΜΕΝΗ ΔΙΑΡΚΕΙΑ ΕΠΙΜΟΡΦΩΤΙΚΟΥ ΣΕΝΑΡΙΟΥ	121
3. ΣΚΟΠΟΙ ΚΑΙ ΣΤΟΧΟΙ ΤΟΥ ΕΠΙΜΟΡΦΩΤΙΚΟΥ ΣΕΝΑΡΙΟΥ	121
4. ΕΠΙΣΤΗΜΟΛΟΓΙΚΗ ΠΡΟΣΕΓΓΙΣΗ ΚΑΙ ΕΝΝΟΙΟΛΟΓΙΚΗ ΑΝΑΛΥΣΗ	122
Στόχοι διδασκαλίας αντικειμενοστρεφούς προγρ/μού	122
Βασικές έννοιες.....	123
Δραστηριότητα 1	124
5. ΔΥΣΚΟΛΙΕΣ, ΠΑΡΑΝΟΗΣΕΙΣ ΚΑΙ ΔΙΔΑΚΤΙΚΕΣ ΠΑΡΕΜΒΑΣΕΙΣ	124
Δραστηριότητα 2	129
Δραστηριότητα 3	129
6. ΔΙΔΑΚΤΙΚΑ ΠΡΟΒΛΗΜΑΤΑ	130

Μετάβαση από τον διαδικαστικό στον αντικειμενοστρεφή προγραμματισμό	131
Η αντικειμενοστραφής τεχνική ανάπτυξης προγραμμάτων είναι δύσκολη	132
Δραστηριότητα 4	133
7. ΧΡΗΣΗ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΛΟΓΙΣΜΙΚΟΥ	133
Γενικές δυσκολίες εκμάθησης του προγραμματισμού	133
Τα πλεονεκτήματα των προγραμματιστικών μικρόκοσμων	135
Μικρόκοσμοι βασισμένοι στο ρομπότ Karel	136
JKarelRobot	137
Jeroo	139
objectKarel	140
Δραστηριότητα 5	141
Εκπαιδευτικά περιβάλλοντα προγραμματισμού	142
8. ΕΝΔΕΙΚΤΙΚΕΣ ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ/ΣΕΝΑΡΙΑ	142
9. ΧΡΗΣΗ ΕΞΩΤΕΡΙΚΩΝ ΠΗΓΩΝ	142
10. ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΠΕΡΑΙΤΕΡΩ ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ – ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΕΡΓΑΣΙΕΣ	145

Προοίμιο

Το επιμορφωτικό υλικό για την επιμόρφωση των εκπαιδευτικών στα Κέντρα Στήριξης Επιμόρφωσης εκπονήθηκε αρχικά στο πλαίσιο της Πράξης «Επιμόρφωση Εκπαιδευτικών στη χρήση και αξιοποίηση των ΤΠΕ στην εκπαιδευτική διδακτική διαδικασία» του ΕΠΕΑΕΚ II, Γ' ΚΠΣ και αφορούσε στους κλάδους εκπαιδευτικών ΠΕ02 (Φιλολογοί), ΠΕ03 (Μαθηματικοί), ΠΕ04 (Φυσικές Επιστήμες) και ΠΕ60/70 (Δάσκαλοι και Νηπιαγωγοί). Στο πλαίσιο της Πράξης «Επιμόρφωση των Εκπαιδευτικών για την Αξιοποίηση και Εφαρμογή των ΤΠΕ στη Διδακτική Πράξη» του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» του ΕΣΠΑ (2007-2013), επικαιροποιήθηκε το παραπάνω υλικό, ενώ εκπονήθηκε το επιμορφωτικό υλικό για τους κλάδους ΠΕ19-20 (καθηγητές Πληροφορικής).

Το επιμορφωτικό υλικό για την επιμόρφωση των εκπαιδευτικών περιλαμβάνει 6 τεύχη: το τεύχος 1, που περιέχει το λεγόμενο Γενικό Μέρος (αφορά όλους τους κλάδους εκπαιδευτικών), και πέντε ακόμη τεύχη του Ειδικού μέρους, ένα για καθένα από τους 5 κλάδους. Το επιμορφωτικό υλικό που αφορά στο ειδικό μέρος της επιμόρφωσης των εκπαιδευτικών ΠΕ19/20 αποτελεί το τεύχος 6, διαιρεμένο στα υποτεύχη 6Α, 6Β, 6Γ.

Το επιμορφωτικό υλικό του παρόντος τεύχους 6Α (τμήμα του οποίου παρήχθη χρησιμοποιώντας ως πρωτογενές υλικό το αντίστοιχο υλικό της εκπαίδευσης επιμορφωτών ΠΕ19/20¹) δημιουργήθηκε από συγγραφική ομάδα, με επικεφαλής τον Καθηγητή του Πανεπιστημίου Μακεδονίας Βασίλειο Δαγδιλέλη και συμμετέχοντες τους εξής:

- Καζανίδη Ιωάννη
- Μαλλιαράκη Χρήστο

¹ Το αντίστοιχο Επιμορφωτικό Υλικό για την εκπαίδευση επιμορφωτών στα Πανεπιστημιακά Κέντρα Επιμόρφωσης (ΠΑΚΕ) δημιουργήθηκε από συγγραφική ομάδα αποτελούμενη από την Καθηγήτρια του ΕΚΠΑ Μαρία Γρηγοριάδου, τον Καθηγητή του Πανεπιστημίου Μακεδονίας Βασίλειο Δαγδιλέλη, τον Καθηγητή του Πανεπιστημίου Πατρών Βασίλειο Κόμη και τον Αναπληρωτή Καθηγητή του Πανεπιστημίου Πελοποννήσου Αθανάσιο Τζιμογιάννη. Στο υλικό συνεισέφεραν οι Σοφία Αγγέλαινα, Αγορίτσα Γόγουλου, Ευαγγελία Γουλή, Μαρία Γρηγοριάδου, Βασίλειος Δαγδιλέλης, Βασίλειος Κόμης, Αναστάσιος Λαδιάς, Δημήτριος Νικολός, Ηλίας Νίτσος, Στυλιανός Ξυνόγαλος, Γεώργιος Πανσεληνάς, Κυπαρισσία Παπανικολάου, Δημήτριος Ρούσσινος, Ευθύμιος Ταμπούρης, Γραμματική Τσαγκάνου, Παναγιώτης Τσιωτάκης, Νίκος Φαχαντίδης, Στασινή Φράγκου.

- Ξυνόγαλο Στυλιανό
- Πανσεληνά Γεώργιο
- Χατζηφωτεινού Αικατερίνη

Ένα τμήμα του επιμορφωτικού υλικού του Τεύχους 1 (Γενικό Μέρος) χρησιμοποιήθηκε ως πρωτογενές για την παραγωγή του παρόντος τεύχους 6Α. Το υλικό του τεύχους 1 είχε δημιουργηθεί και επικαιροποιηθεί από συγγραφική ομάδα με επικεφαλής τον Καθηγητή του Πανεπιστημίου Μακεδονίας Βασίλειο Δαγδιλέλη και συνεργάτη τον Παπαδόπουλο Ιωάννη, Λέκτορα του Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης².

Το επιμορφωτικό υλικό αυτό συντάχθηκε στο πλαίσιο των αρμοδιοτήτων της ειδικής επιστημονικής επιτροπής, αποτελούμενης από τους:

- Χαράλαμπος Ζαγούρα, Καθηγητή του Πανεπιστημίου Πατρών, ο οποίος έχει την ευθύνη συντονισμού των εργασιών της επιτροπής
- Βασίλειο Δαγδιλέλη, Καθηγητή Πανεπιστημίου Μακεδονίας
- Βασίλειο Κόμη, Καθηγητή Πανεπιστημίου Πατρών
- Δημήτριο Κουτσογιάννη, Αναπληρωτή Καθηγητή Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης
- Χρόνη Κυνηγό, Καθηγητή Πανεπιστημίου Αθηνών
- Δημήτριο Ψύλλο, Καθηγητή Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης

Η ως άνω ειδική επιστημονική επιτροπή λειτουργεί στην παρούσα Πράξη ως Επιστημονική Επιτροπή του ΙΤΥΕ-Διόφαντος, συμπράττοντος επιστημονικού φορέα υλοποίησης του έργου.

Το επιμορφωτικό υλικό για την επιμόρφωση των εκπαιδευτικών αποτελεί ιδιοκτησία του ΥπΔΒΜΘ και καλύπτεται από την ισχύουσα νομοθεσία για την προστασία των πνευματικών δικαιωμάτων των δημιουργών.

² Για το Γενικό Μέρος του Επιμορφωτικού Υλικού για τα ΠΑΚΕ συνεργάστηκαν και οι Καψάλης Αχιλλέας, πρώην Καθηγητής στο Πανεπιστήμιο Μακεδονίας, Παπαδόπουλος Ιωάννης, Λέκτορας του Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης, Φαχαντίδης Νικόλαος, Αναπληρωτής Καθηγητής στο Πανεπιστήμιο Δυτικής Μακεδονίας.

ΘΕΩΡΙΑ ΔΙΔΑΚΤΙΚΗΣ ΤΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ενότητα ΘΕΩΡΙΑ ΔΙΔΑΚΤΙΚΗΣ ΤΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ 1

Εισαγωγή στη Διδακτική της Πληροφορικής – διδασκαλία του προγραμματισμού

1. ΣΥΝΟΠΤΙΚΗ ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΕΠΙΜΟΡΦΩΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Η διδασκαλία της Πληροφορικής αποδείχτηκε περίπλοκη στη διδακτική πράξη. Πολλοί μαθητές και σπουδαστές αντιμετωπίζουν προβλήματα κατανόησης εννοιών, παρανόησης των χρησιμοποιούμενων τεχνικών, δυσκολίες στην επιλογή της κατάλληλης μεθόδου όταν καλούνται να αντιμετωπίσουν μια προβληματική κατάσταση. Η διαπίστωση αυτή κατέστησε αναγκαία τη συστηματική διερεύνηση των θεμάτων που σχετίζονται με τη διδασκαλία/εκμάθηση τα Πληροφορικής, έτσι ώστε τελικά η διδασκαλία της να καταστεί πιο αποτελεσματική, δηλαδή οι μαθητές ή σπουδαστές να μαθαίνουν «καλύτερα». Στην παρούσα ενότητα θα παρουσιαστούν μερικές έννοιες από τη Διδακτική της Πληροφορικής, τον κλάδο δηλαδή που ερευνά τα θέματα που σχετίζονται με τη διδασκαλία της Πληροφορικής.

Έμφαση θα δοθεί στη διδασκαλία του προγραμματισμού – ενός κλάδου της Πληροφορικής που έχει προφανώς κεντρική σημασία.

Στη συνέχεια, βασισμένο πάνω στη θεωρία αυτή, θα δοθεί ένα πλαίσιο για μια ιδιαίτερη κατηγορία λογισμικών που χρησιμοποιούνται στη διδασκαλία, τους λεγόμενους *προγραμματιστικούς μικρόκοσμους*.

Στο τελευταίο τμήμα παρουσιάζονται ορισμένοι προγραμματιστικοί μικρόκοσμοι και τρόποι με τους οποίους μπορούν να χρησιμοποιηθούν στη διδασκαλία.

2. ΕΚΤΙΜΩΜΕΝΗ ΔΙΑΡΚΕΙΑ ΕΠΙΜΟΡΦΩΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Προβλέπεται να διαρκέσει συνολικά 3-4 διδακτικές ώρες.

3. ΣΚΟΠΟΙ ΚΑΙ ΣΤΟΧΟΙ ΤΟΥ ΕΠΙΜΟΡΦΩΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Οι επιμορφούμενοι στο τέλος της ενότητας θα πρέπει:

- Να έχουν βασικές γνώσεις των κυριότερων εννοιών της Διδακτικής της Πληροφορικής και να είναι σε θέση να αναγνωρίζουν τις έννοιες αυτές όταν τις συναντήσουν μέσα σε

συγκεκριμένα (διδακτικά) παραδείγματα

- Να γνωρίζουν τις βασικές παραδοχές των μικρόκοσμων προγραμματισμού και των προβληματικών που τους διέπουν
- Να γνωρίζουν μερικούς μικρόκοσμους προγραμματισμού, έτσι ώστε να είναι σε θέση να τους αναζητήσουν (για παράδειγμα στο Διαδίκτυο), να έχουν μια πρώτη, μικρή εμπειρία από τουλάχιστον έναν από αυτούς και να έχουν μια στοιχειώδη γνώση των χαρακτηριστικών τους (για παράδειγμα να γνωρίζουν αν ένας συγκεκριμένος Ο μικρόκοσμος είναι Logo-like).

4. ΘΕΩΡΗΤΙΚΕΣ ΕΝΝΟΙΕΣ ΑΠΟ ΤΗ ΔΙΔΑΚΤΙΚΗ ΤΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

One of our greatest needs in research ideas on mathematical learning and thinking is for conceptual, theory-building analyses of the assumptions we are using in our research (Kilpatrick 1981, p. 370)

Η ανάγκη για ένα θεωρητικό πλαίσιο

Οι έρευνες σχετικά με τη διδασκαλία και την εκμάθηση της Πληροφορικής - και ιδιαίτερα για τον προγραμματισμό των ΗΥ - έχουν γνωρίσει μια σημαντικότερη ανάπτυξη τα τελευταία 30-40 χρόνια. Οι έρευνες αυτές δεν έχουν βέβαια γνωρίσει μια σύμμετρη ανάπτυξη, δηλαδή όλα τα σχετικά αντικείμενα δεν έχουν ερευνηθεί στον ίδιο βαθμό: για παράδειγμα ο προγραμματισμός των ΗΥ έχει ίσως τη μερίδα του λέοντος στο σύνολο των ερευνητικών δραστηριοτήτων κι οι έρευνες που σχετίζονται με τη διδασκαλία εισαγωγικών εννοιών στην Πληροφορική (τα λεγόμενα προβλήματα του *αλφαριθμητισμού στην Πληροφορική* ή τον *προγραμματισμό*) είναι πολύ περισσότερες των υπολοίπων. Υπενθυμίζουμε ότι έχουν ερευνηθεί κυρίως αντικείμενα όπως:

- τα προβλήματα της διδασκαλίας και της κατανόησης των μεταβλητών,
- οι δομές επανάληψης (βρόχοι),
- οι εντολές επιλογής (εντολές τύπου if then else κλπ) όπως η LISP ή λογικού προγραμματισμού όπως η Prolog κλπ σε επιτακτικές γλώσσες προγραμματισμού (π.χ. Pascal, σε αντιδιαστολή με γλώσσες συναρτησιακού προγραμματισμού),
- η αναδρομικότητα κυρίως σε γλώσσες στις οποίες η αναδρομικότητα είναι κυρίαρχη (όπως η LOGO).

Εκτός από τα παραπάνω, έχουν διερευνηθεί επίσης κάποια ιδιαίτερα θέματα όπως η «μεταφορά γνώσεων» από ένα προγραμματιστικό περιβάλλον σε άλλο, η εισαγωγή στον λογικό προγραμματισμό ή τον παράλληλο προγραμματισμό. Σε μεγάλο βαθμό, όπως είναι εύλογο, η έρευνα στη Διδακτική ακολούθησε το «κύμα» των προγραμματιστικών μοντέλων

που είναι ή υπήρξαν κυρίαρχα. Για παράδειγμα, σήμερα, ένα πολύ μεγάλο τμήμα των σχετικών ερευνών σχετίζεται με τη διδασκαλία του αντικειμενοστραφούς προγραμματισμού. Αντίθετα, άλλα θέματα όπως για παράδειγμα η διδασκαλία του δομημένου προγραμματισμού και μάλιστα σε επίπεδο υψηλότερο από της δευτεροβάθμιας εκπαίδευσης, έχουν διερευνηθεί πολύ λιγότερο. Οι λόγοι των ανισόρροπης αυτής κατανομής δεν πρόκειται να εξεταστούν εδώ, απλώς επισημαίνονται.

Στο σύνολο των σχετικών εργασιών ωστόσο, παρά το μεγάλο εύρος της θεματικής, υπάρχει ένας κοινός παρονομαστής: μια ανεπάρκεια θεωρητικού πλαισίου.

Με τον όρο **θεωρητικό πλαίσιο** εννοούμε ένα πλαίσιο αναφοράς, ένα σύνολο υποθέσεων δηλαδή, μέσα στο οποίο οι προτάσεις μιας ερευνητικής εργασίας (οι υποθέσεις εργασίας, οι αναλύσεις, τ' αποτελέσματα και το πεδίο ισχύος των) αποκτούν νόημα σε σχέση μ' αυτό που αποτελεί το ιδιαίτερο αντικείμενο της έρευνας: τα φαινόμενα δηλαδή της διδασκαλίας κι εκμάθησης των προγραμματιστικών εννοιών και του προγραμματισμού γενικότερα, σ' ένα περιβάλλον ειδικά οργανωμένο για το σκοπό αυτό (το εκπαιδευτικό σύστημα).

Το μικρό απόσπασμα του J. Kilpatrick που παρατίθεται ως έξεργο στην αρχή της παραγράφου, συνοψίζει την κατάσταση που επικρατεί στη διδακτική των Μαθηματικών, αλλά την παραθέσαμε γιατί εξίσου καλά εκφράζει την ουσιαστικότερη ανάγκη που υφίσταται σήμερα και στη διδακτική της Πληροφορικής.

Ένα εύλογο ερώτημα είναι το εξής: όσοι διδάσκουν Πληροφορική και προγραμματισμό, για ποιο λόγο χρειάζονται ένα θεωρητικό πλαίσιο; Τι χρειάζεται ένα θεωρητικό πλαίσιο Διδακτικής για να διδάξει κάποιος τις επαναληπτικές δομές ή την κληρονομικότητα;

Το θεωρητικό πλαίσιο επιτρέπει τη νοηματοδότηση των διδακτικών φαινομένων και άρα τον καλύτερο έλεγχο της διδασκαλίας.

Ένα, ακραίο είναι η αλήθεια, παράδειγμα είναι το πέσιμο των μήλων: πριν το Νεύτωνα στα μάτια των ανθρώπων το πέσιμο τους δεν είχε εξήγηση. Τα μήλα απλώς πέφταν (και μάλιστα κάτω από τη μηλιά). Μετά όμως, χάρη στη θεωρία του, γνωρίζουμε ότι η Γη τα έλκει – και μάλιστα ότι και τα μήλα έλκουν τη Γη. Η θεωρία του Νεύτωνα επέτρεψε όχι μόνο να κατανοήσουμε την πτώση των μήλων, αλλά και όλων σωμάτων, σε όλο το Σύμπαν, να γίνουν προβλέψεις φαινομένων, να κατασκευαστούν δορυφόροι κλπ.

Βέβαια η Διδακτική της Πληροφορικής, δεν έχει τέτοιες φιλοδοξίες. Μπορεί όμως να θέσει ένα πλαίσιο για την κατανόηση των δυσκολιών των μαθητών, την αντιμετώπιση των συστηματικών λαθών τους και την πιο αποτελεσματική

διδασκαλία: για παράδειγμα ο εκπαιδευτικός να γνωρίζει εκ των προτέρων τα πιθανά εμπόδια που θα συναντήσουν οι μαθητές και να προετοιμάσει κατάλληλα το μάθημα.

Δυο προκαταρκτικές παρατηρήσεις επιβάλλονται:

α) θα πρέπει ίσως να καταστήσουμε σαφές το γεγονός ότι η Διδακτική (της Πληροφορικής, αλλά και των Μαθηματικών, της Φυσικής κλπ) διακρίνεται ως ξεχωριστός κλάδος από μια σειρά άλλων κλάδων όπως η παιδαγωγική, η γνωστική ψυχολογία, η επιστημολογία κλπ κατά τούτο: η διδακτική όχι μόνο επικεντρώνεται ιδιαίτερα στα προβλήματα της διδασκαλίας αλλά και κάνει αποκλειστικές αναφορές *στο προς διδασκαλία αντικείμενο*, δηλαδή θεωρεί τα σχετικά φαινόμενα στο πλαίσιο της ιδιαίτερου επιστημονικού κλάδου στον οποίο αναφέρεται (στην περίπτωση μας της Πληροφορικής κι ιδιαίτερα του προγραμματισμού).

β) μια δεύτερη παρατήρηση αφορά στην έννοια της "θεωρίας": η θεωρία εξυπονεί ένα συναφές και μη αντιφατικό εννοιολογικό πλαίσιο το οποίο αποσκοπεί στην ερμηνεία και πρόβλεψη φαινομένων σχετικών μ' ένα συγκεκριμένο πεδίο.

Η θεωρία πρέπει βέβαια να συνδέεται με ειδικά και σαφώς καθορισμένα προβλήματα και κατ' αυτήν την έννοια ίσως ο καταλληλότερος όρος για την περιγραφή του πλαισίου της Διδακτικής στο οποίο κάνουμε αναφορά είναι η "**προβληματική**". Μια προβληματική είναι ευρύτερη από την θεωρία, παραπέμπει σ' αυτήν αλλά δεν συνοψίζεται σ' αυτήν.

Για να δώσουμε ένα παράδειγμα, αν ένας ερευνητής προτείνει σε μια ομάδα μαθητών μια σειρά από ερωτήσεις ώστε να εξακριβώσει τις αντιλήψεις τους τις σχετικές με την έννοια της μεταβλητής, οφείλει ν' αποσαφηνίσει και να διευκρινίσει τις υποθέσεις που κάνει:

- ότι δηλαδή πράγματι υπάρχουν κάποιες ιδιαίτερες αντιλήψεις των μαθητών
- ότι οι απαντήσεις των μαθητών στα συγκεκριμένα ερωτήματα που θέτει θα τον βοηθήσουν στην διερεύνηση των αντιλήψεων αυτών κλπ.

Οι αντιλήψεις των σπουδαστών

Στη μελέτη των διδακτικών φαινομένων που μας απασχολούν, κεντρικό ρόλο παίζουν οι λεγόμενες αντιλήψεις των σπουδαστών: πρόκειται για ένα σύνολο υποθετικών γνώσεων τις οποίες ο ερευνητής αποδίδει στο μαθητή ή το σπουδαστή προκειμένου να περιγράψει και να προβλέψει ορισμένες από τις αντιδράσεις του. Οι αντιλήψεις αυτές μπορούν να έχουν διάφορες πηγές: μπορεί εξίσου καλά να προέρχονται από τον κοινωνικό περίγυρο, την καθημερινή εμπειρία αλλά και τις ίδιες τις διδακτικές πρακτικές κλπ.

Οι αντιλήψεις λοιπόν λειτουργούν ως ένα είδος γνώσεων - αποτελούν στην πραγματικότητα

μια γνώση η οποία καθοδηγεί την πράξη. Οι αντιλήψεις αυτές δεν είναι βέβαια άμεσα παρατηρήσιμες: συνάγουμε την ύπαρξη τους (δηλαδή την θεωρητική ύπαρξη τους) από την παραγωγή των μαθητών και σπουδαστών (τους αλγόριθμους και προγράμματα τους, τις απαντήσεις και τα σχόλια τους κ.ά.). Η ύπαρξη συστηματικών λαθών - λαθών δηλαδή που έχουν έναν επίμονο και διαρκή, γενικευμένο χαρακτήρα - αποδίδεται στη ύπαρξη σχετικών αντιλήψεων.

Το λάθος δηλαδή δεν είναι μόνο το αποτέλεσμα της άγνοιας, της αβεβαιότητας, της τύχης - όπως θεωρούν οι εμπειρικοστικές ή μηχηβιοριστικές θεωρίες μάθησης, αλλά είναι και το αποτέλεσμα μιας προγενέστερης γνώσης, που κάποτε ήταν αποτελεσματική, είχε ενδιαφέρον, αλλά τώρα αποδεικνύεται λανθασμένη ή απλά ακατάλληλη. Τα λάθη αυτού του τύπου δεν είναι απρόβλεπτα, ούτε "πλάνες" - παράγονται από εμπόδια.

Παραθέτουμε ορισμένα παραδείγματα:

Απλοϊκές αντιλήψεις: το παράδειγμα του ανθρωπομορφισμού

Ένα παράδειγμα εδραιωμένης αντίληψης, «κλασσικό» στο είδος του, αντιστοιχεί στην ονομαζόμενη ανθρωπομορφική αντίληψη. Ο σχετικός όρος οφείλεται στον Pea R. D. [1984] και χρησιμοποιείται για να δηλώσει το γεγονός ότι οι αρχάριοι προγραμματιστές πολλές φορές ερμηνεύουν τα προγράμματα και γενικότερα την επικοινωνία τους με ένα πληροφορικό σύστημα όχι με βάση τους συντακτικούς και σημασιολογικούς κανόνες (syntax, semantics) της χρησιμοποιούμενης γλώσσας (προγραμματισμού, γλώσσας εντολών κλπ), αλλ' αποδίδοντας στο αντίστοιχο πρόγραμμα ένα νόημα ευρύτερο του πραγματικού, σαν να πρόκειτο για έναν διάλογο μ' άνθρωπο. Η ανθρωπομορφική αντίληψη έχει διάφορες εκφράσεις, όπως αναφέρει και ο Pea. Ως μια τέτοια εκδήλωση ανθρωπομορφισμού θεωρούμε το γεγονός ότι, κυρίως οι νεαροί μαθητές, έχουν την τάση - τις πρώτες φορές που χρησιμοποιούν έναν τερματικό σταθμό ή έναν μικροϋπολογιστή - να θεωρούν ότι μπορούν να έχουν «διάλογο» με το σύστημα, όπως στον παρακάτω διάλογο (πρόκειται για εντελώς αυθεντικό περιστατικό):

Μαθητής : MY NAME IS GEORGE. WHAT IS YOUR NAME ?

Πληρ. σύστ. : Bad command or missing file

Μαθητής :

Ο όρος ανθρωπομορφική αντίληψη αντιστοιχεί στην ερευνητική υπόθεση ότι η λανθασμένη ερμηνεία του προγράμματος οφείλεται σε μια λανθασμένη αντίληψη (δηλαδή σε μια υποθετική γνώση των μαθητών) που ονομάζεται ανθρωπομορφική γιατί οι μαθητές συμπεριφέρονται σαν το πληροφορικό σύστημα να διέθετε μια αυτόνομη ανθρώπινη νόηση - χωρίς αυτό να σημαίνει βέβαια ότι οι μαθητές πιστεύουν στ' αλήθεια στην ύπαρξη μιας

αυτόνομης νόησης στο «εσωτερικό» του Η.Υ. Απλώς συμπεριφέρονται σαν... Ανάλογη αντίληψη με αυτήν είναι εκείνη κατά την οποία οι νεαροί μαθητές μεταφέρουν στο προγραμματιστικό περιβάλλον στρατηγικές επίλυσης προβλημάτων από άλλα περιβάλλοντα - χωρίς να κάνουν τις απαραίτητες τροποποιήσεις. Αν, για παράδειγμα, προταθεί σε αρχάριους μαθητές το ακόλουθο πρόγραμμα:

...

```
EMVADON:= VASIS * YPSOS
```

```
READLN (VASIS);
```

```
READLN (YPSOS);
```

```
WRITELN (EMVADON);
```

...

και τους ζητηθεί να προβλέψουν το αποτέλεσμα της εκτέλεσης του (αν για παράδειγμα δοθούν οι τιμές 8 και 4 στις μεταβλητές VASIS και YPSOS αντίστοιχα), το σύνολο των μαθητών προβλέπει ως απάντηση τον αριθμό 32 (Pea R. D. [1984], Dagdilelis V. [1986]). Η απάντηση αυτή θα ήταν βέβαια ορθή αν επρόκειτο για μια συνηθισμένη άσκηση σχολικής γεωμετρίας όχι όμως για ένα πρόγραμμα Η.Υ.

Το γεγονός ότι το φαινόμενο αυτό έχει διαπιστωθεί:

- με μαθητές προερχόμενους από τρεις τουλάχιστον διαφορετικές χώρες,
- οι οποίοι έχουν διδαχθεί τα σχετικά αντικείμενα από διαφορετικούς καθηγητές - που χρησιμοποίησαν βέβαια και διαφορετικές μεθόδους διδασκαλίας -

δείχνει ότι το πρόβλημα μοιάζει να προέρχεται από ένα αίτιο ανεξάρτητο της μεθόδου διδασκαλίας. Λαμβάνουν υπ' όψη τους το γεγονός ότι όλα τα δεδομένα του προβλήματος εμπεριέχονται στο πρόγραμμα (ο τύπος του εμβადού και τα μήκη βάσεως και ύψους) και θεωρούν αυτόματα (και βεβαίως λανθασμένα) ότι το πληροφορικό σύστημα (δηλαδή ο ΗΥ και το λογισμικό) θα παράσχει την σωστή απάντηση, χωρίς να λαμβάνουν ταυτόχρονα υπ' όψη τους και το γεγονός ότι η λειτουργία του προγράμματος είναι γραμμική κι όχι «παράλληλη» ή οπισθοδρομούσα (ο Pea προτείνει ακριβώς τον όρο «παράλληλισμός» για να χαρακτηρίσει αυτή την κατηγορία λαθών).

Αντιλήψεις σχετιζόμενες με τη χρήση επαναληπτικών διαδικασιών

Ο ανθρωπομορφισμός αυτός, ας τον ονομάσουμε «απλοϊκό», φαίνεται να εξαφανίζεται σχετικά γρήγορα με την πάροδο του χρόνου - χωρίς ν' απαιτηθεί κάποια ιδιαίτερη διδακτική μέθοδος. Άλλες όμως μορφές λανθασμένης ερμηνείας αλγορίθμων και προγραμμάτων

φαίνονται να είναι περισσότερο επίμονες κι εμφανίζονται και σε λιγότερο αρχάριους σπουδαστές: για παράδειγμα η χρήση των βρόχων και γενικότερα των επαναληπτικών διαδικασιών, παραμένει πολύ συχνά ένα θέμα το οποίο οι σπουδαστές δεν κατανοούν απόλυτα (για περισσότερα παραδείγματα δες τις εξωτερικές πηγές).

Οι περιπτώσεις στις οποίες αναφερόμαστε παραπάνω μπορούν να δώσουν την εντύπωση του περιθωριακού φαινομένου, την εντύπωση δηλαδή ότι δεν παίζουν ουσιαστικό ρόλο στην πορεία του διδασκόμενου. Τούτο όμως δεν αληθεύει γιατί ορισμένες αντιλήψεις των μαθητών - συνδεόμενες για παράδειγμα με τη χρήση των βρόχων - αποτελούν ουσιαστικό ανασταλτικό παράγοντα στην πορεία και η υπέρβαση τους είναι απαραίτητη για την πρόοδο του μαθητή.

Τα είδη των συστηματικών λαθών που αναφέραμε παραπάνω αφορούν σε αρχάριους μαθητές και ένα εύλογο ερώτημα που προκύπτει είναι το ακόλουθο: συστηματικά λάθη ή δυσκολίες εμφανίζονται και σε σπουδαστές λιγότερο αρχάριους; Η απάντηση είναι και πάλι καταφατική.

Η αναδρομή

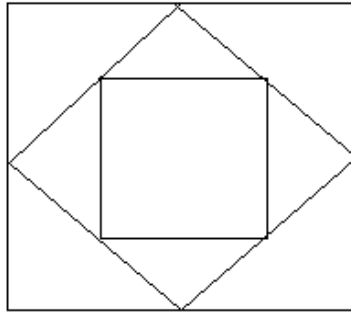
Σ' ένα διαφορετικό επίπεδο αναφέρουμε επίσης τις δυσκολίες που συχνά συναντούν οι σπουδαστές στη χρήση αναδρομικών διαδικασιών. Αν και η αναδρομή αποτελεί ένα σπουδαίο διανοητικό εργαλείο για την αλγοριθμική επίλυση προβλημάτων και χαρακτηρίζεται από την μεγάλη της απλότητα, σύμφωνα τουλάχιστον με τη γνώμη πολλών επιστημόνων (Arsac J. [1983]), ωστόσο η χρήση αναδρομικών διαδικασιών φαίνεται πως είναι εξαιρετικά δυσχερής όχι μόνο για τους νεαρούς μαθητές, αλλά ακόμη και για σπουδαστές πανεπιστημιακού επιπέδου. Το πρόβλημα της κατασκευής αναδρομικών διαδικασιών έχει διερευνηθεί εκτεταμένα σε νεαρά παιδιά σχολικής ηλικίας κι εφήβους (παράδειγμα οι σχετικές μελέτες για τη χρήση της γλώσσας LOGO), αλλά, απ' όσο γνωρίζουμε, έχει ελάχιστα μελετηθεί σε σπουδαστές πέραν της δευτεροβαθμίου εκπαίδευσως (εκτός από μια σειρά ερευνών για τη χρήση της γλώσσας PROLOG). Παρ' όλη την έλλειψη σχετικών ερευνών, είναι ωστόσο γενικά αποδεκτή η παρατήρηση ότι συχνά οι σπουδαστές διαχειρίζονται με λανθασμένο τρόπο τις αναδρομικές διαδικασίες ή αδυνατούν να τις χρησιμοποιήσουν - και τούτο ακόμη κι όταν τα προτεινόμενα προβλήματα έχουν λύσεις που περιγράφονται με προφανή τρόπο σε άλλο περιβάλλον. Ας θεωρήσουμε το παράδειγμα ενός κλασσικού προβλήματος: επιθυμούμε από ένα κείμενο να εξαλείψουμε τα περίσσια κενά (αν υπάρχουν) ανάμεσα στις λέξεις του (μια διαδικασία συνήθης σε επεξεργασία κειμένου). Ενώ η μέθοδος για την εξάλειψη των περίσσιων κενών είναι

προφανής, ωστόσο η αλγοριθμική περιγραφή μιας τέτοιας διαδικασίας είναι λιγότερο τετριμμένη και η περιγραφή μιας αναδρομικής συνάρτησης

$F(\text{κείμενο}, \text{κείμενο})$

η οποία θα δεχόταν ως όρισμα ένα κείμενο και θα έδινε ως αποτέλεσμα το ίδιο κείμενο απαλλαγμένο από τα περίσσια κενά, είναι ακόμη πιο δυσχερής.

Ακόμη, η απευθείας σχεδίαση ή κατασκευή σχημάτων με «αναδρομική» δομή, (δηλαδή με δομή η οποία μπορεί να περιγραφεί αναδρομικά), όπως το παρακάτω:



είναι βέβαια τετριμμένη, αλλ' αντίθετα η περιγραφή μιας αναδρομικής διαδικασίας για την σχεδίαση τους από ένα πληροφορικό σύστημα είναι λιγότερο τετριμμένη.

Τα σχετικά λοιπόν θέματα θα πρέπει, κατά τεκμήριο, να θεωρηθούν «δύσκολα» για τους σπουδαστές.

Οι ακαδημαϊκοί και οι επαγγελματίες προγραμματιστές: η σχέση με τα «λάθη»

Η τελευταία κατηγορία προγραμματιστών για τους οποίους τίθεται το ερώτημα των «δυσκολιών» είναι οι έμπειροι προγραμματιστές. Υπάρχουν συστηματικά λάθη στους επαγγελματίες προγραμματιστές; Μπορούν τα λάθη αυτά ν' αποδοθούν σε λανθασμένες αντιλήψεις των προγραμματιστών; Η απάντηση είναι και πάλι καταφατική - αν και τούτο μοιάζει με παράδοξο: πώς είναι δυνατό οι πεπειραμένοι προγραμματιστές να κάνουν συστηματικά λάθη; Πώς είναι δυνατό να εκφράσουν λανθασμένες αντιλήψεις πεπειραμένοι προγραμματιστές;

Αρχικά είναι βέβαια γνωστό ότι, γενικά, το παραγόμενο λογισμικό όχι μόνο δεν είναι απαλλαγμένο λαθών αλλ' επιπλέον η εκσφαλμάτωση του (debugging) αποτελεί έναν εξαιρετικό δύσκολο τομέα κι ασφαλώς έναν από τους δαπανηρότερους στην παραγωγή λογισμικού (αργότερα θα έχουμε την ευκαιρία να αναφερθούμε εκτενέστερα στο θέμα). Η ανάλυση των λαθών (για παράδειγμα στο εισαγωγικό μέρος του Beizer B. [1983]) δείχνει ότι η λανθασμένη χρήση βρόχων ή η λανθασμένη-ελλιπής απόδοση αρχικών τιμών σε μεταβλητές (initializing) αποτελούν συνήθη σφάλματα στους επαγγελματίες προγραμματιστές.

Πέραν όμως της γενικής αυτής θεώρησης των σφαλμάτων, θα μπορούσαν να αναφερθούν και συγκεκριμένα παραδείγματα προβλημάτων η λύση των οποίων συχνά χαρακτηρίζεται

από λάθη. Μερικά τέτοια παραδείγματα δίνονται παρακάτω:

1. Το πρόβλημα της δυαδικής αναζήτησης (binary search).

Πρόκειται για ένα διάσημο πρόβλημα, τετριμμένο, τουλάχιστον εκ πρώτης όψεως, και μεγέθους που δεν ξεπερνά τις 10-15 γραμμές. Παρ' όλ' αυτά η πολυπλοκότητα της λύσεώς του είναι μεγάλη. Ο τρόπος με τον οποίο ο προγραμματιστής αναπαριστά νοητά την λειτουργία του αλγόριθμου παίζει σπουδαίο ρόλο στην επιλογή των *αναλλοίωτων σχέσεων* (invariances) κι επομένως στην κατασκευή του αλγόριθμου (δες για παράδειγμα Paradaikos Nikos [1988]). Ο Lesuisse R. [1983] αναλύοντας πολλούς δημοσιευμένους αλγορίθμους δυαδικής αναζήτησης σε έγκυρα επιστημονικά περιοδικά, επισημαίνει το γεγονός ότι οι αλγόριθμοι κατασκευάστηκαν σε πολλές περιπτώσεις με βάση μια νοητή αναπαράσταση της λειτουργίας τους, με αποτέλεσμα ορισμένοι απ' αυτούς να εμφανίζουν μια σειρά ατελειών και λαθών. Ο Lesuisse κατατάσσει τους αλγόριθμους σε τέσσερις κατηγορίες και κάνει την υπόθεση ότι πηγή των λαθών είναι τελικά ο τρόπος της σύλληψης και κατασκευής του αλγόριθμου - με άλλα λόγια τα λάθη κι οι ασάφειες δεν είναι τυχαία αλλά η πηγή τους μπορεί να προσδιοριστεί στον ίδιο τον τρόπο κατασκευής τους.

2. Δυο προβλήματα επεξεργασίας κειμένου

Το πρώτο αναφέρεται σε μια μικρή διαδικασία (procedure) η οποία όταν δεχθεί ως όρισμα ένα κείμενο επιστρέφει ως αποτέλεσμα ένα κείμενο ευθυγραμμισμένο δεξιά. Ο αλγόριθμος προτάθηκε μαζί με την απόδειξη του από τον Naur P. [1972 b] και μάλιστα ως υπόδειγμα αποδεικτικής μεθοδολογίας, αλλά όπως δείχνει ο Lesuisse R. [1984], ο αλγόριθμος περιέχει λάθη. Ως πηγή των λαθών ο Lesuisse R. θεωρεί την λανθασμένη αντιμετώπιση του προβλήματος από τον Naur P. και πιο συγκεκριμένα στο γεγονός ότι ο Naur P. αντιμετώπισε μ' εσφαλμένο τρόπο τα ίδια τ' αντικείμενα τα οποία επεξεργαζόταν ο αλγόριθμος του (δηλαδή λέξεις, παραγράφους, προτάσεις). Το σημείο αυτό είναι σημαντικό και για τη δική μας εργασία, διότι ενισχύει την υπόθεση ότι η σχεδίαση ορθών αλγορίθμων δεν σημαίνει απλά την εφαρμογή ορισμένων τεχνικών, αλλά απαιτεί μια πολύ ριζικότερη μεταβολή στην αντιμετώπιση των προβλημάτων. Στο σημείο αυτό -θα επανέλθουμε αργότερα.

Το δεύτερο παράδειγμα είναι ένα πρόβλημα επινοημένο από τους Henderson P., Snowdon R. [1972] ακριβώς για να χρησιμοποιηθεί ως παράδειγμα στη διδασκαλία του δομημένου προγραμματισμού (structured programming). Ζητείται η κατασκευή ενός αλγόριθμου για τη μέτρηση του πλήθους των λέξεων που υπάρχουν σε μια σειρά τηλεγραφημάτων (telegram flow analysis). Παρά τη φαινομενική απλότητα του προβλήματος, υπήρξαν μια σειρά δημοσιευμένες λύσεις - που περιείχαν λάθη όμως - από έμπειρους προγραμματιστές (αναφέρουμε ενδεικτικά τον Ledgard H. [1973]). Οι ιδιαιτερότητες του αλγορίθμου τον έχουν καταστήσει «κλασικό» για τη διδασκαλία του δομημένου προγραμματισμού. Θα αναφερθούμε ιδιαίτερα στο πρόβλημα αυτό, διότι το χρησιμοποιήσαμε στα πειράματα που

πραγματοποιήσαμε με σπουδαστές.

3. Λάθη, δοκιμές και σύγχρονη τεχνολογία λογισμικού

Ορισμένοι πληροφορικοί, έχοντας υιοθετήσει μια οπτική γωνία πλησιέστερα στις ανάγκες του επαγγελματία προγραμματιστή παρά στις ανάγκες του ακαδημαϊκού ερευνητή (θ' αναφερθούμε λεπτομερέστερα στο θέμα σε κατοπινές παραγράφους) θεωρούν ότι οι μοντέρνες μεθοδολογίες για την επικύρωση αλγορίθμων δεν εξασφαλίζουν οπωσδήποτε τη δημιουργία πιο αξιόπιστων αλγορίθμων κι εκθέτουν μια σειρά περιπτώσεων δημοσιευμένων αλγορίθμων οι οποίοι περιέχουν ασάφειες, λάθη, παραλείψεις κι εξαιρέσεις που δεν μνημονεύονται.

4. Ένας αλγόριθμος "αποκομιδής απορριμμάτων" (garbage collection)

Τέλος, αναφέρουμε και το παράδειγμα ενός αλγορίθμου "αποκομιδής απορριμμάτων" ο οποίος δημοσιεύτηκε στο έγκυρο περιοδικό CACM και μάλιστα βραβεύθηκε, αλλά περιείχε λάθη (αναφέρεται από τον Gries D. [1977]).

Επιχειρώντας να διερευνήσουμε τα πιθανά αίτια, τις "πηγές" ορισμένων αντιλήψεων των μαθητών και σπουδαστών μπορούμε να διαπιστώσουμε τα εξής:

α) Η αντίληψη του παραλληλισμού την οποία αναφέραμε, μπορεί να οφείλεται στην ελλιπή κατανόηση των αρχών λειτουργίας των πληροφορικών συστημάτων κι ιδιαίτερα στο γεγονός ότι τα προγράμματα - και μάλιστα τα μικροσκοπικά προγράμματα που συνήθως κατασκευάζουν οι σπουδαστές - εκτελούνται πολύ γρήγορα: *it looks at the program all at once because it is so fast* σχολίασε ένας σπουδαστής όταν ρωτήθηκε σε μια σχετική έρευνα (Pea [1984], σελ. 4).

β) Η απλοϊκή μορφή του ανθρωπομορφισμού (ο μαθητής που προσπαθεί να μάθει τ' όνομα του ΗΥ) μπορεί να βρίσκει την πηγή της στην περιρρέουσα ατμόσφαιρα της καθημερινής ζωής στην οποία οι ΗΥ (παλιότερα "ηλεκτρονικοί εγκέφαλοι") εμφανίζονται να πραγματοποιούν ολοένα και πιο εντυπωσιακά επιτεύγματα και άρα θα μπορούσε κανείς να συμπεράνει ότι οι ΗΥ διαθέτουν ένα είδος «ευφυΐας».

γ) Το λάθος κατά το οποίο οι μαθητές αποδίδουν τιμές χωρίς να λαμβάνουν υπόψη τους τον τρόπο με τον οποίο λειτουργεί το σύστημα οφείλεται ίσως στην εμπειρία από άλλα μαθήματα στα οποία βέβαια η σειρά με την οποία περιγράφονται τα δεδομένα (μήκος, πλάτος, σχετικός μαθηματικός τύπος) δεν παίζουν ουσιαστικό ρόλο στην επίλυση ενός προβλήματος υπολογισμού εμβαδού.

Οι πιθανές πηγές προέλευσης λοιπόν των αντιλήψεων αυτών μπορούν να είναι ποικίλες. Ορισμένες από τις αντιλήψεις αυτές - μολονότι λανθασμένες στη γενική τους έκφραση -

έχουν κάποιο πεδίο εφαρμογών γεγονός που τις κάνει να είναι πιο "ανθεκτικές" σε μια ενδεχόμενη τροποποίηση τους. Η δημιουργία κι η εδραίωση μιας αντίληψης οφείλεται στο ότι - τουλάχιστον σε ορισμένες περιπτώσεις - η αντίληψη αυτή είναι λειτουργική. Αν για παράδειγμα σε κάποια είδη προβλημάτων η αντίληψη ενός σπουδαστή αποτελεί εργαλείο για την εξεύρεση μιας λύσης, τότε βέβαια η σχετική αντίληψη εδραιώνεται κι η τροποποίηση της θα είναι ακόμη πιο δύσκολη.

Στην περίπτωση αυτή η λανθασμένη αντίληψη μπορεί ν' αποτελέσει ένα πραγματικό **εμπόδιο** στην πρόοδο του.

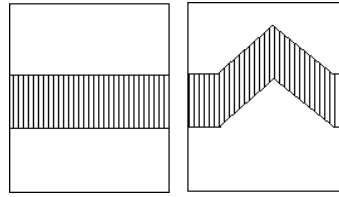
Μια καινούρια γνώση στο σχολικό περιβάλλον (με τη γενική έννοια του όρου) θεσμοθετείται από τη στιγμή που από τον καθηγητή προσδιορίζεται ρητά. Ωστόσο είναι προφανές ότι η κατανόηση των νέων εννοιών δεν ακολουθεί κατ' ανάγκη τη σειρά και τους ρυθμούς της διδασκαλίας. Μια καινούρια έννοια αποκτά νόημα ανάλογα με τις καταστάσεις στις οποίες λειτουργήσει, ανάλογα με τα προβλήματα που λύθηκαν χάρη σε αυτήν. Οι γνώσεις των μαθητών, έτσι όπως εξελίσσονται μέσα στο χρόνο, σπάνια ακολουθούν ένα γραμμικό μοντέλο ανάπτυξης - δηλαδή συνεχούς προοδευτικής πορείας μέχρι ένα επιθυμητό σημείο. Πολύ συχνά βρίσκονται σε αντίθεση με το διδακτικό περιβάλλον (φυσικός χώρος για τη γεωμετρία, προσομοιούμενη πραγματικότητα αν πρόκειται για προσομοίωση σε ΗΥ, θεωρητική αν πρόκειται για μια αξιωματικοποιημένη δομή) και καμιά φορά υφίστανται ταυτοχρόνως αντιλήψεις ασύμβατες κι αντιφατικές. Το εμπόδιο προβάλλει λοιπόν όχι ως μια έλλειψη γνώσης, αλλά ως μια γνώση που δεν είναι πια κατάλληλη. Το εμπόδιο έχει μερικά χαρακτηριστικά:

- πρόκειται για μια γνώση η οποία λειτουργεί αποτελεσματικά σε μια σειρά καταστάσεων και για ορισμένες τιμές των διδακτικών μεταβλητών (δες παρακάτω).
- το εμπόδιο είναι μια γνώση η οποία όταν πρόκειται να προσαρμοσθεί σε άλλες καταστάσεις - ή σε άλλες τιμές των διδακτικών μεταβλητών - προκαλεί ιδιαίτερα λάθη, αναγνωρίσιμα κι αναλύσιμα ως προς το εμπόδιο
- το εμπόδιο - δηλαδή αυτή η ιδιαίτερη γνώση - ανθίσταται στις μετατροπές.
- το εμπόδιο λοιπόν δεν μπορεί να ξεπεραστεί παρά μόνο χάρη σε ιδιαίτερες καταστάσεις απόρριψης.

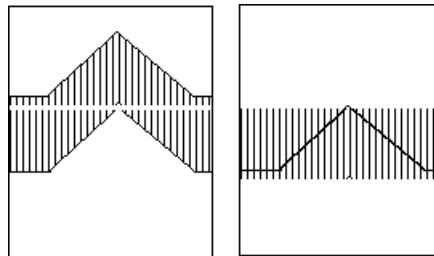
Αν η αντίληψη αυτή (η αντίστοιχη γνώση) έχει μια συχνή χρήση μπορεί ακόμη να μετεξελιχθεί ένα είδος **διαίσθησης**, ένα είδος δηλαδή άμεσης και εννοιατικής γνώσης που τίθεται σε εφαρμογή κατά τρόπο σχεδόν αυτόματο.

Χωρίς να επεκταθούμε στην περί διαίσθησης θέση του Fishbein, αναφέρουμε το εξής χαρακτηριστικό παράδειγμα:

Δίνονται τ' ακόλουθα δύο σχήματα κι ερωτάται ποιο απ' τα δυο γραμμοσκιασμένα μέρη έχει μεγαλύτερο εμβαδόν:



Ο Fishbein (1983) αποδίδει σε μια διαίσθηση το γεγονός ότι οι σπουδαστές θεωρούν ότι το γραμμοσκιασμένο μέρος του δευτέρου σχήματος έχει μεγαλύτερο εμβαδόν. Βέβαια η θεώρηση αυτή εύκολα μπορεί ν' αποδειχθεί ψευδής:



Αυτό όμως δεν συνεπάγεται την εξαφάνιση της λανθασμένης αντίληψης ότι “όσο ένα σχήμα έχει μεγαλύτερη περίμετρο, τόσο έχει και μεγαλύτερο εμβαδόν”. Η αντίληψη αυτή είναι ισχυρή κι όπως δείχνουν σχετικές έρευνες (ibid) αποτελεί σχεδόν μια διαίσθηση - ο ερωτώμενος απαντά σχεδόν αυτόματα κι είναι πεπεισμένος για την ορθότητα της απάντησης του. Ακόμη και χαρακτηριστικά παραδείγματα δεν είναι επαρκή για να τροποποιήσουν την αντίληψη αυτή. Για παράδειγμα οι μαθητές αποδέχονται εύκολα το γεγονός ότι η Ελλάδα ενώ έχει κατά πολύ μικρότερο εμβαδόν από την Αφρικανική Ήπειρο έχει σχεδόν το ίδιο μήκος ακτών με αυτή. Τούτο όμως δεν τους εμποδίζει να εφαρμόζουν στη συνέχεια - για παράδειγμα στο μάθημα της γεωμετρίας - την ίδια λανθασμένη αντίληψη ότι μεγαλύτερο εμβαδόν αντιστοιχεί σε μεγαλύτερη περίμετρο

Οι λανθασμένες αντιλήψεις, όπως τουλάχιστον δείχνουν οι σχετικές έρευνες, μπορούν ακόμη να συνυπάρχουν και μ' άλλες γνώσεις (αντιλήψεις) οι οποίες να είναι ασύμβατες μ' αυτές ή ακόμη και αντιφατικές και να μην εμφανίζονται παρά μόνο κάτω από ειδικές συνθήκες. Η ύπαρξη τους πάντως μπορεί να διαπιστωθεί από την συστηματική εμφάνιση τους, όταν πληρούνται οι κατάλληλες προϋποθέσεις. Οι αντιλήψεις των σπουδαστών οι σχετικές μ' ένα αντικείμενο πριν από τη διδασκαλία και μετά απ' αυτήν αποτελούν βέβαια ένα σημαντικότερο θέμα για τη διδακτική της Πληροφορικής. Εξάλλου ένα σημαντικό μεθοδολογικό πρόβλημα της διδακτικής είναι ακριβώς η σχεδίαση, πραγματοποίηση κι

ανάλυση τέτοιων καταστάσεων οι οποίες επιτρέπουν την εμφάνιση συγκεκριμένων αντιλήψεων των σπουδαστών.

Οι αντιλήψεις των σπουδαστών, εφ' όσον αποτελούν ένα είδος γνώσης, μπορούν να έχουν διαφορετικούς τρόπους έκφρασης. Έτσι μια αντίληψη μπορεί να εκφραστεί σαν αλγόριθμος ή σαν ένα είδος *έμπρακτου θεωρήματος* (théorème-en-acte) το οποίο δεν εκφράζεται ρητά αλλά εφαρμόζεται στην πράξη. Έτσι, αν για παράδειγμα μια συγκεκριμένη διαδικασία πρόκειται να εκτελεστεί ένα συγκεκριμένο πλήθος φορών, τότε η καταλληλότερη μορφή βρόχου είναι του τύπου FOR...DO...NEXT. Τούτο έχει και μια δευτερεύουσα συνέπεια: οι σπουδαστές οι οποίοι είναι ιδιαίτερα εξοικειωμένοι μ' ένα στυλ προγραμματισμού (π.χ. με μια ορισμένη κλάση προβλημάτων ή με μια ορισμένη γλώσσα προγραμματισμού) μπορεί με ορισμένα έμπρακτα θεωρήματα περιορισμένης χρηστικότητας τα οποία βέβαια μπορούν ν' αποτελέσουν ανασταλτικούς παράγοντες σε περίπτωση εργασίας σε ένα άλλο προγραμματιστικό περιβάλλον.

Συνοψίζοντας θα μπορούσαμε ν' αναφέρουμε ότι οι αντιλήψεις των μαθητών και των σπουδαστών δεν είναι άμεσα παρατηρήσιμες, αλλά μόνον έμμεσα, αναλύοντας τον τρόπο με τον οποίο αντιδρούν στα πλαίσια ορισμένων καταστάσεων. Η διδακτική στοχεύει κατ' αρχήν στη διαπίστωση των αντιλήψεων αυτών, προκειμένου να επιτύχει τη βασική της επιδίωξη: την μετατροπή των αντιλήψεων αυτών σε άλλες λειτουργικότερες και πιο αποτελεσματικές - γεγονός που συνήθως περιγράφεται ως *απόκτηση γνώσεων*.

Η διδασκόμενη γνώση κι ο μετασχηματισμός της

Μια βασική θέση της διδακτικής είναι ότι *η διδασκόμενη γνώση δεν ταυτίζεται με την αντίστοιχη επιστημονική γνώση*. Για παράδειγμα η αναδρομικότητα, ή ακριβέστερα οι αναδρομικοί αλγόριθμοι, ως επιστημονική έννοια έχουν διερευνηθεί σε βάθος από τους μαθηματικούς κι έχουν χρησιμοποιηθεί ευρύτατα από τους πληροφορικούς. Ωστόσο οι αληθινές τους διαστάσεις στο επαγγελματικό πεδίο και στην επιστημονική τους χρήση δύσκολα μπορούν να παρουσιαστούν μέσα στο σύνθητες εκπαιδευτικό περιβάλλον.

Συνήθως μια έννοια δεν μπορεί να περιγραφεί πλήρως κι ακόμη περισσότερο να γίνει κατανοητή, από έναν απλό τυπικό ορισμό. Κατά κανόνα μία έννοια αποκτά το πλήρες νόημα της μέσα σ' ένα πλαίσιο αναφοράς το οποίο περιλαμβάνει τις συγγενείς προς αυτήν έννοιες, τα προβλήματα τα οποία μπορεί να επιλύσει - όπως κι εκείνα που δεν μπορεί - κλπ. Το πλαίσιο αυτό ονομάζεται **εννοιολογικό πεδίο** (champ conceptuel).

Αυτό που συνήθως παρουσιάζεται σε μια εκπαιδευτική διαδικασία είναι ένα τμήμα του εννοιολογικού πεδίου, εκείνο το τμήμα που ικανοποιεί τις τρέχουσες διδακτικές ανάγκες. Σε

ορισμένες περιπτώσεις η μερική αυτή παρουσίαση μπορεί ακόμη και να στρεβλώσει το επιστημονικό νόημα της αντίστοιχης εννοίας. Χρησιμοποιώντας ένα παράδειγμα από τα Μαθηματικά, η έννοια της απόστασης έχει μια τρέχουσα διδακτική χρήση αρκετά διαφορετική από την μαθηματικά ορισμένη. Ένα ακόμη πιο χαρακτηριστικό παράδειγμα αποτελούν οι δεκαδικοί αριθμοί των οποίων η εισαγωγή όχι μόνο δεν έχει καμία σχέση με την αναγκαιότητα χρήσης τους, αλλά αποτελεί πιθανότατα και το βασικό αίτιο για την δημιουργία λανθασμένων αντιλήψεων στους νεαρούς μαθητές.

Επιπλέον, αν μια περιγραφή της προς διδασκαλία γνώσης μπορεί να βρεθεί στα διδακτικά εγχειρίδια και τις σημειώσεις, θα πρέπει να σημειωθεί ότι η περιγραφή αυτή, κατά κανόνα, δεν αναφέρει τον λόγο ύπαρξης της γνώσης αυτής. Η Πληροφορική αποσκοπεί, όπως όλες οι επιστήμες, στην επίλυση προβλημάτων. Τα προβλήματα αυτά μπορούν να προέρχονται από την ίδια την εξέλιξη της επιστήμης, να είναι "εσωτερικά" ή να βρίσκουν την προέλευση τους σε άλλες πηγές (τεχνική, επαγγελματικές ανάγκες, άλλες επιστήμες...). Τα εννοιολογικά εργαλεία κάθε επιστήμης επιτρέπουν την επίλυση ακριβώς των προβλημάτων αυτών. Το να δοκιμάσει λοιπόν κανείς να μεταδώσει μια γνώση χωρίς να καταστήσει γνωστά τα προβλήματα τα οποία μπορούν να επιλυθούν χάρη σ' αυτήν, είναι σαν μια απάντηση σε μια ερώτηση που δεν τέθηκε ποτέ.

Μια γνώση, έτσι όπως εκφράζεται σε μια δεδομένη στιγμή, αποτελεί επίσης το αποτέλεσμα απ' όλες τις διαμάχες που προηγήθηκαν για την εξέλιξη της. Έτσι, για παράδειγμα οι τρέχουσες απόψεις για τον δομημένο προγραμματισμό και την χρήση των διαφόρων "jumps" (exit, goto κλπ) αποτελούν το προϊόν όλων των συζητήσεων και διαμαχών που προηγήθηκαν.

Είναι προφανές ότι όλη αυτή η προβληματική δεν μπορεί να μεταφερθεί αυτούσια στη σχολική ή την πανεπιστημιακή αίθουσα.

Αντίθετα, οι νέες έννοιες που εισάγονται στην εκπαιδευτική διαδικασία υπακούουν σε διαφορετικούς κανόνες:

- ✓ παρουσιάζονται κατά κανόνα σε συνδυασμό με προβλήματα ορισμένου τύπου και με ορισμένη μορφή
- ✓ παρουσιάζονται με τρόπο τέτοιο ώστε ν' ανταποκρίνονται στις (υποτιθέμενες) τρέχουσες γνώσεις των σπουδαστών
- ✓ κατακερματίζονται σύμφωνα με τις διδακτικές ανάγκες: στα βιβλία και τις σημειώσεις χωρίζονται σε παραγράφους, σε ενότητες διδάξιμες σε περίοδο ακεραίου αριθμού εξαμήνων ή ετών.

Για όλους τους παραπάνω λόγους το "νόημα" της σχολικής γνώσης θα είναι διαφορετικό από

το "πραγματικό" νόημα της, αφού οι συνθήκες δημιουργίας και λειτουργίας της είναι διαφορετικές στις δυο περιπτώσεις. Η γνώση, έτσι όπως εμφανίζεται μέσα στην εκπαιδευτική διαδικασία αποτελεί την τελική εικόνα μιας σειράς μετασχηματισμών, οι οποίοι έχουν την ονομασία **διδακτικός μετασχηματισμός** ή **διδακτική μετατόπιση** (transposition didactique).

Οι επιστημονικές γνώσεις λοιπόν, σπάνια διδάσκονται αυτές καθαυτές στις διάφορες σχολικές βαθμίδες. Το γεγονός αυτό είναι μάλιστα πιο σαφές στις χαμηλότερες εκπαιδευτικές βαθμίδες. Επί της ουσίας, η μόνη εκπαιδευτική βαθμίδα όπου η επιστημονική γνώση αποτελεί αντικείμενο διδασκαλίας είναι ο χώρος μέσα στον οποίο παράγεται, δηλαδή το πανεπιστήμιο. Αντίθετα, στις άλλες βαθμίδες της εκπαίδευσης διδάσκεται ένα σώμα γνώσης, το οποίο χαρακτηρίζεται από τον όρο «σχολικές γνώσεις», οι οποίες προκύπτουν κατά κανόνα από τις επιστημονικές γνώσεις μιας περιοχής ως προϊόν μιας διαδικασίας μετασχηματισμού. Ο διδακτικός μετασχηματισμός χρησιμοποιήθηκε για πρώτη φορά στο πλαίσιο της Διδακτικής των Μαθηματικών για να καταδειχθεί αυτή τη διαδικασία. Πρέπει να τονισθεί ότι η έννοια του διδακτικού μετασχηματισμού είχε ήδη δημιουργηθεί από τον κοινωνιολόγο Michel Verret το 1975.

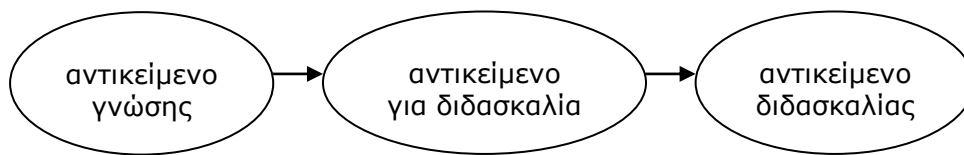
Με την έννοια του διδακτικού μετασχηματισμού οριοθετούνται οι γενικοί μηχανισμοί που επιτρέπουν το πέρασμα από ένα «αντικείμενο επιστημονικής γνώσης» σε ένα «αντικείμενο διδασκαλίας». Ξεκινώντας από τη διάκριση ανάμεσα στην «επιστημονική γνώση» (όπως αυτή παράγεται από την επιστημονική κοινότητα στο πλαίσιο της επιστημονικής έρευνας) και στη «διδαχθείσα γνώση» (όπως αυτή μπορεί να παρατηρηθεί στην καθημερινή σχολική πρακτική) η Διδακτική μελετά πως γίνεται ο **μετασχηματισμός** των επιστημονικών εννοιών ώστε να καταστούν αντικείμενο διδασκαλίας.

Στο πλαίσιο αυτό, ένα περιεχόμενο γνώσης έχοντας ορισθεί ως διδακτέα γνώση, υπόκειται σε ένα σύνολο από προσαρμοστικούς μετασχηματισμούς που θα το καταστήσουν ικανό να πάρει θέση ανάμεσα στα αντικείμενα διδασκαλίας. Συνεπώς, η διανοητική «εργασία» η οποία μετατρέπει ένα αντικείμενο γνώσης για διδασκαλία σε αντικείμενο διδασκαλίας, αποκαλείται διδακτικός μετασχηματισμός.

Ο διδακτικός μετασχηματισμός δεν περιγράφει ωστόσο μόνο τη μετατροπή μιας επιστημονικής γνώσης σε διδακτική γνώση με τη μορφή κεφαλαίων σχολικών βιβλίων για παράδειγμα. Διαπνέει, αντιθέτως, όλο το φάσμα της διδακτικής πράξης και είναι σε στενή σχέση με τον τόπο, το κοινό και τους παιδαγωγικούς και διδακτικούς στόχους που τίθενται. Επηρεάζει συνεπώς ουσιαστικά το πρόγραμμα σπουδών, τα αναλυτικά προγράμματα και τη διδακτική πρακτική.

Η έννοια του διδακτικού μετασχηματισμού επιτρέπει να χαρακτηρίσουμε το μετασχηματισμό της γνώσης ανάμεσα σε μια **κατάσταση αναφοράς** και μια **κατάσταση μετάδοσης**. Η κατάσταση αναφοράς αφορά αυτό που θεωρείται (σε κάποια χρονική περίοδο) επιστημονική γνώση όπως αυτή παράγεται από τους επιστήμονες. Η κατάσταση μετάδοσης αφορά όλο το σχολικό σύστημα, αποτέλεσμα του οποίου είναι η σχολική γνώση αλλά και η γνώση που έχει προσκτηθεί από τους μαθητές (Κόμης, 2001).

Όπως είδαμε, ο διδακτικός μετασχηματισμός αφορά καταρχήν μια σειρά από μετασχηματισμούς (δες παρακάτω σχήμα) στους οποίους συμμετέχουν τόσο οι δημιουργοί αναλυτικών προγραμμάτων όσο και οι συγγραφείς σχολικών βιβλίων, οι σχολικοί σύμβουλοι, κλπ.).



Σταδιακή μετατροπή της γνώσης από επιστημονική σε «σχολική»: διδακτική μετατόπιση

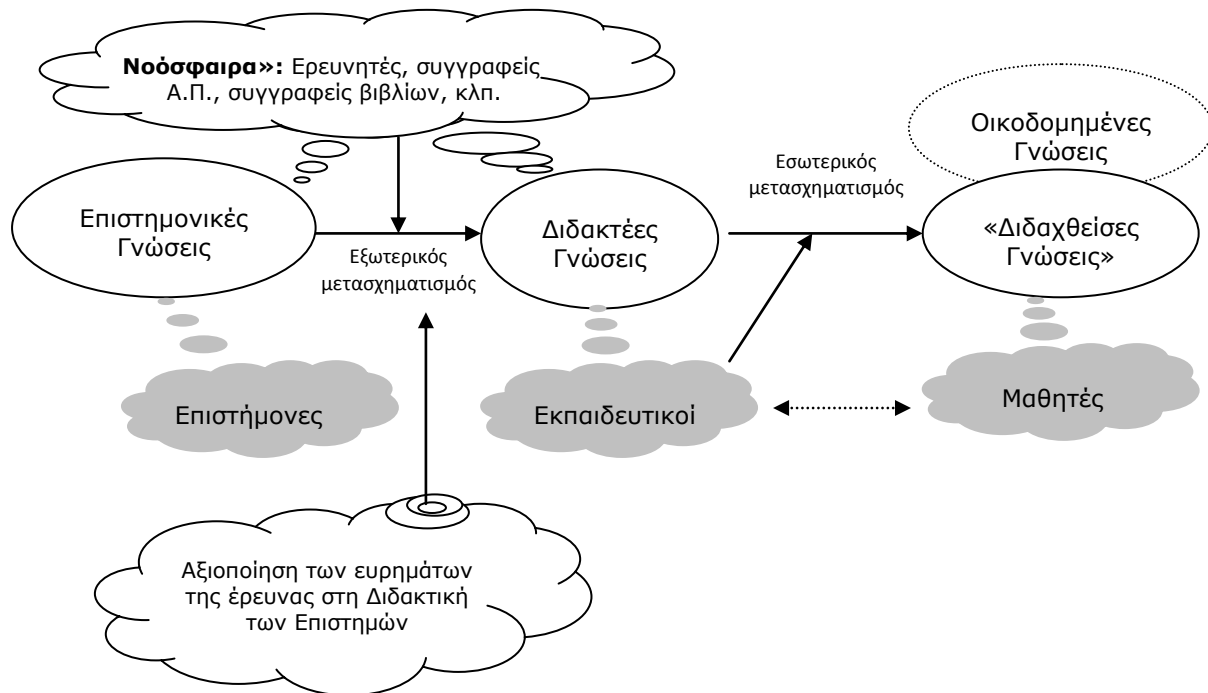
Το «αντικείμενο γνώσης» προσδιορίζεται συνεπώς από το χώρο της «επιστημονικής γνώσης», αυτό δηλαδή που αναγνωρίζεται από την επιστημονική κοινότητα ως τέτοιο. Εάν πάρουμε για παράδειγμα τα «Λειτουργικά Συστήματα» στην Πληροφορική, θα δούμε ότι το αντικείμενο αυτό δεν είναι διδάξιμο, αυτό καθαυτό, τουλάχιστον εκτός του χώρου παραγωγής του, δηλαδή τα Πανεπιστήμια και τα Ινστιτούτα.

Συγκεκριμένοι μηχανισμοί πρέπει να τεθούν σε λειτουργία ώστε η εν λόγω γνώση (όπως είναι καταγεγραμμένη στις επιστημονικές δημοσιεύσεις και στα ακαδημαϊκά εγχειρίδια) να βγει από τον επιστημονικό της χώρο και να εισαχθεί λειτουργικά και αποτελεσματικά στη διδακτική πράξη. Από τη στιγμή που πραγματοποιηθούν τέτοιου τύπου λειτουργίες, η διδακτέα γνώση είναι αναμφισβήτητα διαφορετική από την επιστημονική γνώση που χρησιμεύει ως αναφορά της. Πρέπει επίσης να τονισθεί ότι το επιστημολογικό της περιβάλλον είναι επίσης διαφορετικό όπως και η σημασία της και η εμβέλεια των εννοιών που την δομούν.

Η έννοια του διδακτικού μετασχηματισμού συνεπώς υπονοεί ότι το πέρασμα από την επιστημονική γνώση στη διδακτέα γνώση δεν είναι ποτέ άμεσο. Επιπλέον, ο καθορισμός μιας επιστημονικής γνώσης σε αντικείμενο διδασκαλίας τροποποιεί σε μεγάλο βαθμό τη φύση της γνώσης αυτής αφού αφενός αλλάζουν τα ερωτήματα που μπορούν να απαντηθούν μέσω αυτής της γνώσης αλλά και οι σχέσεις που η ίδια διατηρεί με άλλες.

Ο διδακτικός μετασχηματισμός μπορεί στο πλαίσιο αυτό να θεωρηθεί ως μια διαδικασία με

δύο κύρια στάδια (δες παρακάτω σχήμα) και οι μετασχηματισμοί που πραγματοποιούνται μπορούν να κατηγοριοποιηθούν σε δύο ομάδες: σε ένα εξωτερικό μετασχηματισμό και σε ένα εσωτερικό μετασχηματισμό.



Τα στάδια του διδακτικού μετασχηματισμού

Οι ιδιαίτερες προσαρμογές που πραγματοποιεί κάθε εκπαιδευτικός γίνονται κατά το δεύτερο στάδιο του διδακτικού μετασχηματισμού. Συνεπώς ο εκπαιδευτικός δεν παίζει ρόλο παρά στο δεύτερο στάδιο και οι αρμοδιότητές του κατά το μετασχηματισμό είναι ιδιαίτερα περιορισμένες. Ο διδακτικός μετασχηματισμός, γίνεται σε μεγάλο βαθμό έξω από αυτόν.

Αντίθετα, το πρώτο στάδιο συνιστά αυτό που αποκαλείται στη Διδακτική (με μια δόση ειρωνείας είναι η αλήθεια) «**νοόςφαιρα**». Με τον όρο «νοόςφαιρα» εννοείται ο χώρος όλων αυτών που προγραμματίζουν τα περιεχόμενα της διδασκαλίας, δημιουργούν τα αναλυτικά προγράμματα, συγγράφουν τα σχολικά εγχειρίδια, κλπ. Η νοόςφαιρα παίζει το ρόλο του ενδιάμεσου ανάμεσα στο σύστημα διδασκαλίας και στο ευρύτερο κοινωνικό περιβάλλον.

Βέβαια, ακόμα κι αν οι εκπαιδευτικοί είναι ανίσχυροι να επέμβουν στα πλαίσια της νοόςφαιρας, μπορούν να τροποποιήσουν το «κείμενο της γνώσης»: κάτω από το πρίσμα αυτό, ο διδακτικός μετασχηματισμός συνίσταται για τον εκπαιδευτικό στο να οικοδομήσει τα μαθήματά του αντλώντας από τις επιστημονικές γνώσεις, παίρνοντας υπόψη του τους

προσανατολισμούς και τις οδηγίες των αναλυτικών προγραμμάτων (διδασκόμενες γνώσεις) ώστε να τις προσαρμόσει στην τάξη του. Ο μετασχηματισμός επιβάλλει την εξαγωγή μιας γνώσης από το πλαίσιο της (πανεπιστημιακό, κοινωνικό, κλπ.) και την επανένταξή της μέσα στο ιδιαίτερο πλαίσιο της τάξης.

Πρέπει στο σημείο αυτό να τονιστεί ότι τα αντικείμενα που θεωρούνται ως «αντικείμενα διδασκαλίας» δεν προέρχονται σε καμία περίπτωση από «απλουστεύσεις» πιο σύνθετων αντικειμένων που προέρχονται από την επιστημονική κοινότητα. Είναι, αντιθέτως, το αποτέλεσμα μιας διδακτικής προσπάθειας, μιας αποσύνθεσης, μιας κατασκευής, που τα κάνει να διαφέρουν ποιοτικά.

Ο όρος του *μετασχηματισμού* υπογραμμίζει ότι η γνώση δεν μπορεί να μεταδοθεί αυτή καθαυτή και ο όρος του *διδακτικού* καθιστά εμφανή την αναγκαιότητα εύρεσης κανόνων αυτού του μετασχηματισμού που να είναι κατάλληλοι στη δομή της γνώσης που πρέπει να προσκτηθεί και τρόπους που λαμβάνουν υπόψη τους τη διαδικασία της μάθησης.

Το διδακτικό συμβόλαιο

Ορισμένες φορές οι μαθητές, αλλά κι οι σπουδαστές, δίνουν απαντήσεις που φαίνονται παράλογες. Το πιο "κλασικό" παράδειγμα τέτοιου παραλογισμού αποτελεί το γνωστό πρόβλημα της ηλικίας του καπετάνιου: σ' ένα πλοίο υπάρχουν 26 πρόβατα και 10 κατσίκες. Πόσων χρόνων είναι ο καπετάνιος; Η συντριπτική πλειοψηφία των μαθητών 8-9 ετών απαντά ότι ο καπετάνιος είναι 36 ετών. Δύσκολα θα μπορούσαμε ν' αποδώσουμε τον παραλογισμό αυτό σ' έλλειψη γνώσεων ή πνευματικών ικανοτήτων - προφανώς οι μαθητές της 3ης δημοτικού σχολείου έχουν τ' απαραίτητα εφόδια ώστε ν' απαντήσουν σωστά. Ορισμένοι ερευνητές έδειξαν ότι στην πραγματικότητα μια τέτοια συμπεριφορά μπορεί να εξηγηθεί μέσα σ' ένα ευρύτερο πλαίσιο, γιατί η απάντηση των μαθητών μπορεί να είναι παράλογη με την κοινή λογική αλλά είναι απολύτως συμβατή με τη συνήθη σχολική λογική. Τα σχολικά προβλήματα υπακούουν στους εξής κανόνες:

α) ένα συνηθισμένο πρόβλημα έχει μια και μοναδική απάντηση

β) για την απάντηση αυτή:

όλα τα δεδομένα του προβλήματος πρέπει να χρησιμοποιηθούν
καμιά άλλη ένδειξη ή δεδομένο δεν είναι απαραίτητη

Η σωστή χρήση των δεδομένων γίνεται χάρη σ' ένα τυποποιημένο σχήμα εφαρμογής οικείων διαδικασιών, οι οποίες πρέπει απλά να συνδυαστούν με τον κατάλληλο τρόπο.

Τέλος το αποτέλεσμα πρέπει να υπακούει σε ορισμένους κανόνες κοινής λογικής: σ' ένα πρόβλημα με αυγά η απάντηση δεν πρέπει να είναι "2,3 αυγά", μια ηλικία ανθρώπου δεν

μπορεί να είναι 200 ετών κλπ.

Τα συνηθισμένα προβλήματα στη Δευτεροβάθμια εκπαίδευση επιπέδου δεν χαρακτηρίζονται βέβαια από ένα τόσο τυποποιημένο σύστημα - όμως ορισμένα από τα χαρακτηριστικά αυτά μπορούν να εντοπιστούν και στο επίπεδο της.

Μ' έναν γενικό τρόπο δεχόμαστε την ύπαρξη ενός συνόλου άγραφων, έμμεσων αλλά υπαρκτών και ισχυρών κανόνων οι οποίοι προσδιορίζουν ορισμένα χαρακτηριστικά των προβλημάτων που προτείνονται μέσα στα εκπαιδευτικά πλαίσια αλλά και γενικότερα των αμοιβαίων "νομίμων" συμπεριφορών τόσο του καθηγητή όσο και των μαθητών και σπουδαστών. Το σύνολο των κανόνων αυτών ονομάζεται **διδακτικό συμβόλαιο**.

Το διδακτικό συμβόλαιο χαρακτηρίζει κατά μεγάλο μέρος τις αλληλεπιδράσεις μεταξύ σπουδαστών και καθηγητή. Χαρακτηρίζει επίσης το είδος των προβλημάτων τα οποία είναι "νόμιμα". Για παράδειγμα, την περίοδο της πρώτης επαφής με τον προγραμματισμό ένα βασικό πρόβλημα που τίθεται στους αρχαρίους είναι ακριβώς ο προσδιορισμός της φύσης των ζητούμενων: ενώ συνήθως η λύση των προβλημάτων έγκειται στον προσδιορισμό κάποιων αριθμών, αλγεβρικών παραστάσεων ή γεωμετρικών σχημάτων, στον προγραμματισμό αυτό που συνήθως ζητείται είναι μια μέθοδος επίλυσης (ένα πρόγραμμα δεν επιλύει ένα πρόβλημα, αλλά μια κλάση προβλημάτων, περιγράφει δηλαδή μια μέθοδο επίλυσης). Έτσι πολλές φορές οι σπουδαστές χωρίς μεγάλη εμπειρία στον προγραμματισμό, δυσκολεύονται να κατανοήσουν τη διαφορά ανάμεσα στο παράδειγμα και τη μέθοδο. Αν προταθεί ως πρόβλημα ο προσδιορισμός ενός τρόπου για την αντιμετάθεση των περιεχομένων δύο μεταβλητών:

για παράδειγμα αν $A=3$ και $B=5$ τότε επιθυμούμε μετά την εφαρμογή της διαδικασίας να έχουμε $B=5$ και $A=3$, πολλοί μαθητές αλλά και σπουδαστές προτείνουν σα λύση την εκχώρηση

$B:=5$

$A:=3$

Ίσως θεωρεί κανείς ότι στην τριτοβάθμια εκπαίδευση τα φαινόμενα αυτά εξαλείφονται γρήγορα, δεδομένου ότι οι σπουδαστές "σκέφτονται" πιο ορθολογικά - κατά κάποιο τρόπο το "συμβόλαιο" τείνει να παίζει έναν ρόλο ασήμαντο. Παρά ταύτα το είδος των ερωτήσεων που τίθενται, για παράδειγμα κατά τη διάρκεια των εξετάσεων, συχνά μαρτυρά ότι οι σπουδαστές δείχνουν μια τάση προσαρμογής στις υποτιθέμενες προθέσεις του καθηγητή παρά στο ρητά ζητούμενο που εκφράζει η εκφώνηση: δεν είναι για παράδειγμα σπάνιο φαινόμενο αν ζητηθεί ένα τμήμα αλγορίθμου ή ένας αλγόριθμος από ένα ευρύτερο σύνολο μεθόδων, οι σπουδαστές ν' απαντήσουν δίνοντας πολύ περισσότερα απ' όσα ζητούνται.

Στα πλαίσια αυτά θεωρούμε επίσης ότι μπορεί να υφίσταται μια αντιδιαστολή ανάμεσα σ' αυτό που ονομάζουμε **ιδιωτική πραγματικότητα** και στην **δημόσια εικόνα**. Όπως κι αλλού έχουμε αναφέρει, η προσωπική, ιδιωτική εργασία ενός επιστήμονα μπορεί να είναι - και κατά κανόνα είναι - τελείως διαφορετική από την τελική μορφή των αποτελεσμάτων που δημοσιοποιούνται. Κατά μία τελείως ανάλογη έννοια η προσωπική εργασία ενός μαθητή ή σπουδαστή, τα στοιχεία εκείνα που του χρησιμεύουν ως δείκτης για την πορεία του, που αποτελούν για τον ίδιο μια επιβεβαίωση της ορθότητας των επιλογών του, μπορεί να είναι τελείως διαφορετικά από τα στοιχεία που παρουσιάζει ως τελική εργασία. Για παράδειγμα, στις εξετάσεις η προσωπική του εργασία (που συνήθως αντικατοπτρίζεται σ' αυτό που ονομάζουμε *πρόχειρο*) μπορεί να είναι τελείως διαφορετική από την τελική, επίσημη μορφή του γραπτού (το *καθαρό*). Το πρώτο αντιστοιχεί περισσότερο στις προσωπικές του επιλογές, αποτελεί ένα δείγμα του τι πιστεύει ο ίδιος για την εργασία του, ενώ το δεύτερο κατ' ανάγκη είναι προσαρμοσμένο στις απαιτήσεις του διδακτικού συμβολαίου.

Η βασική υπόθεση της Διδακτικής

Η βασική υπόθεση της Διδακτικής είναι διπλή:

α) το υποκείμενο (ο σπουδαστής) μαθαίνει - δηλαδή αυξάνει και τροποποιεί τις γνώσεις του - χάρη στην αλληλεπίδραση του με το περιβάλλον του. Η αλληλεπίδραση αυτή στο εκπαιδευτικό σύστημα έγκειται βασικά σε μια δραστηριότητα επίλυσης προβλημάτων σε συνθήκες που προσδιορίζει, μερικώς τουλάχιστον, ο εκπαιδευτής-καθηγητής. Η διδασκαλία δεν είναι τίποτε άλλο παρά η δημιουργία ενός περιβάλλοντος το οποίο αποσκοπεί στο να μετασχηματίσει τις αντιλήψεις των σπουδαστών. Η διδακτική έχει σα στόχο της την ανάλυση των διαδικασιών αυτών και γενικότερα την εξήγηση κι ερμηνεία των διδακτικών φαινομένων.

β) Βασική επιστημολογική υπόθεση της Διδακτικής είναι ότι τα προβλήματα είναι *πηγή και κριτήριο της γνώσης* (Vergnaud 1982, σ. 220). Από τη μια η γνώση πηγάζει από προβλήματα, η γνώση κατά κανόνα δημιουργείται προκειμένου να δοθούν λύσεις σε προβλήματα κι απ' την άλλη μια διανοητική παραγωγή αποτελεί γνώση στο βαθμό που είναι σημαντική, που επιτρέπει τη λύση μιας κλάσης προβλημάτων με αποτελεσματικό τρόπο.

Κεντρική έννοια των υποθέσεων αυτών είναι το πρόβλημα. Για να είναι δυνατή η μάθηση πρέπει οι σπουδαστές να διαθέτουν κάποιους τρόπους επίλυσης του προτεινόμενου προβλήματος. Η μάθηση συντελείται όταν μέσα από την επίλυση του προβλήματος ο σπουδαστής τροποποιεί τους τρόπους αυτούς, τους βελτιώνει, δημιουργεί νέους,

υποβιβάζει την αξία παλαιότερων. Σε ορισμένες περιπτώσεις ο έλεγχος αυτός μπορεί να πραγματοποιηθεί μέσα από τροποποιήσεις του ίδιου του προβλήματος. Όταν οι τροποποιήσεις αυτές παίζουν ένα ρόλο στην μάθηση ονομάζονται **διδακτικές μεταβλητές**.

Ορισμένες μεταβλητές είναι εγγενείς στο πρόβλημα. Αν για παράδειγμα ένα παιδί μπορεί να μετρήσει μέχρι το δέκα με τα δάχτυλα, τότε η απαρίθμηση ενός συνόλου 20 στοιχείων προφανώς ελαττώνει την αξία της μεθόδου αυτής. Η επιλογή του πληθικού αριθμού του προς απαρίθμηση συνόλου είναι πολύ σημαντική γιατί αν η τιμή του είναι "γειτονική" με το 10, τα παιδί θα τείνει να προσαρμόσει την ήδη γνωστή διαδικασία (με τα δάχτυλα) κι επομένως η διαδικασία αυτή θα έχει όλες τις πιθανότητες να "επιζήσει" μετά τη δοκιμασία, ενώ ένας πολύ μεγάλος πληθικός αριθμός μπορεί να "μπλοκάρει" τελείως το παιδί.

Αναφερόμενοι στον προγραμματισμό, μπορούμε να διαπιστώσουμε την ύπαρξη διδακτικών μεταβλητών. Προκειμένου να επιδειχθούν τα προτερήματα του δομημένου προγραμματισμού, προφανώς καταλληλότερο είναι ένα σχετικά πολύπλοκο και μεγάλο πρόβλημα, αφού οι μέθοδοι της δοκιμής-λάθους ή της απευθείας επίλυσης είναι τελείως αναποτελεσματικές στις περιπτώσεις αυτές - ενώ είναι λιγότερο δαπανηρές σε απλά προβλήματα. Το μέγεθος κι η πολυπλοκότητα του προτεινομένου προβλήματος είναι ουσιαστικοί παράγοντες στην περίπτωση αυτή. Η ανάγκη σχεδίασης κι ελέγχου αλγορίθμων μπορεί αντίθετα να επιτευχθεί με ρύθμιση μεταβλητών που είναι εξωγενείς ως προς το πρόβλημα: μπορεί για παράδειγμα οι σπουδαστές να έχουν ορισμένο μόνο χρόνο μηχανής για την κατασκευή των προγραμμάτων τους, γεγονός που θα τους αναγκάσει να τελειοποιήσουν τους αλγορίθμους τους και να τους ελέγξουν προσεκτικά πριν περάσουν στην πληκτρολόγηση του τελικού κώδικα - ενώ αντίθετα η πιο συνηθισμένη τακτική είναι η προσφυγή στην άμεση πληκτρολόγηση κώδικα ο οποίος διορθώνεται ανάλογα με τα λάθη που διαπιστώνονται. Η δυνατότητα ή όχι χρήσης εντολών όπως η goto ή ορισμένου τύπου jumps μπορεί επίσης σε συνδυασμό με την κατάλληλη επιλογή προβλημάτων να παίξει έναν ουσιαστικό ρόλο στην κατανόηση κι αποτελεσματική χρήση των διαφόρων τύπων βρόχων.

Η Διδακτική ως πειραματική επιστήμη

Το πρόβλημα στην εκπαιδευτική διαδικασία, όπως δείχνουν και τα παραπάνω, δεν πρέπει να νοηθεί σαν μια "εκτός πλαισίου", context-free πρόταση.

Τα προβλήματα πρέπει μάλλον να θεωρούνται σαν μια πρόταση η οποία πρέπει να επιλυθεί μέσα σε ορισμένα πλαίσια - και τα πλαίσια αυτά ρυθμίζουν κατά τρόπο ουσιαστικό τις διαδικασίες που είναι θεμιτές, "νόμιμες" για την επίλυση του προβλήματος, καθώς κι εκείνες που είναι αποτελεσματικές. Τα προβλήματα προς επίλυση μέσα στα πλαίσια αυτά

ονομάζονται **διδακτικές καταστάσεις**.

Σε μια διδακτική κατάσταση όλοι εκείνοι οι παράγοντες οι οποίοι ανταποκρίνονται σε μια ορισμένη ενέργεια του υποκειμένου και γίνονται αντιληπτοί απ' αυτό ονομάζονται **περιβάλλον**.

Για παράδειγμα ο καθηγητής (ο οποίος αντιδρά μ' έναν ορισμένο τρόπο σε μια απάντηση ενός σπουδαστή), ο συνάδελφος ο οποίος αποδέχεται ή απορρίπτει ένα επιχείρημα ή μια απόδειξη, το πληροφορικό σύστημα που παράγει ένα αποτέλεσμα ή ένα μήνυμα κατά την εκτέλεση ενός προγράμματος, η ηλεκτρονική μηχανή τσέπης αποτελούν μέρη του περιβάλλοντος. Είναι αυτονόητο πως η εκάστοτε "αντίδραση" του περιβάλλοντος συντελεί στη διαδικασία της μάθησης κι η δημιουργία του πλέον κατάλληλου περιβάλλοντος αποτελεί βασικό μέλημα της Διδακτικής.

Πολλοί ερευνητές διακρίνουν διαφόρους τύπους διδακτικών καταστάσεων όπως:

καταστάσεις ενέργειας στις οποίες ευνοείται κυρίως η κατασκευαστική ικανότητα των μαθητών γι' απλά αντικείμενα όπως ένα γεωμετρικό σχήμα

καταστάσεις έκφρασης ή διατύπωσης οι οποίες ευνοούν την κατασκευή κι εκμάθηση μοντέλων και επιστημονικών "γλωσσών". Ιδιαίτερο ρόλο ανάμεσα τους παίζουν οι *καταστάσεις επικοινωνίας*, στις οποίες η επικοινωνία (δηλαδή μια διατύπωση για τη μετάδοση ενός μηνύματος) παίζει καθοριστικό ρόλο (Laborde [1982]): αν η ποιότητα μιας διατύπωσης δεν επιτρέπει την ακριβή μετάδοση ενός μηνύματος, τότε επανεξετάζεται όχι μόνο η ίδια η διατύπωση κι η χρησιμοποιούμενη "γλώσσα" αλλά και η υποκείμενη γνώση

καταστάσεις επικύρωσης οι οποίες έχουν ως επίκεντρο την παραγωγή επιχειρημάτων για την επικύρωση μιας σχέσης, μιας πρότασης μιας κατασκευής (για παράδειγμα ενός αλγόριθμου ή ενός προγράμματος). Σε ορισμένες περιπτώσεις οι *καταστάσεις απόφασης* (Balacheff [1991]) απαιτούν ένα ορισμένο όριο βεβαιότητας, χωρίς ωστόσο να είναι απαραίτητες οι αποδείξεις (δηλαδή η ρητή διατύπωση εκείνων των επιχειρημάτων που δικαιολογούν την βεβαιότητα αυτή).

καταστάσεις επισημοποίησης οι οποίες υφίστανται υπό την ευθύνη του καθηγητή κι αποσκοπούν στο να δώσουν το χαρακτήρα της θεσμικά αναγνωρισμένης γνώσης, σε ορισμένες από τις προσωπικές γνώσεις του σπουδαστή.

Ένα μεγάλο τμήμα της έρευνας της Διδακτικής είναι επιστημολογικού χαρακτήρα: αναζήτηση των πηγών και της εξέλιξης των γνώσεων, των εννοιών που πρόκειται να διδαχθούν. Ένα τμήμα της περιλαμβάνει επίσης την επεξεργασία ενός θεωρητικού πλαισίου, ή ακριβέστερα μιας προβληματικής όπως το περιγράψαμε παραπάνω. Το σημαντικότερο όμως τμήμα της είναι καθαρά πειραματικού χαρακτήρα: η Διδακτική είναι μια κατεξοχήν πειραματική

επιστήμη. Με βάση τις προηγούμενες (θεωρητικές και επιστημολογικές) αναλύσεις προσδιορίζεται ένα σύνολο *a priori* αντιδράσεων των σπουδαστών και στη συνέχεια η υλοποίηση της σχετικής διδακτικής κατάστασης επιτρέπει την αντιπαραβολή των όσων είχαν προβλεφθεί σε σχέση με τα όσα πράγματι συνέβησαν - προκειμένου να προσαρμοσθεί κατάλληλα η όλη σύλληψη.

Θα πρέπει ίσως να τονιστεί ότι στην τρέχουσα μεθοδολογία της Διδακτικής (ακριβέστερα: στην κυρίαρχη ευρωπαϊκή σχολή Διδακτικής) επικρατεί ως μέθοδος ανάλυσης η ποιοτική κι όχι η ποσοτική ανάλυση. Τούτο γιατί η Διδακτική δεν ενδιαφέρεται τόσο για την κατανομή των διαφόρων φαινομένων μέσα στον μαθητικό ή σπουδαστικό πληθυσμό αλλά μάλλον για τα αίτια που προκαλούν τα φαινόμενα αυτά καθώς και για τους πιθανούς τρόπους μεταβολής τους, ενδιαφέρεται δηλαδή περισσότερο για ποιοτικά παρά για ποσοτικά χαρακτηριστικά.

Μια τυπική έρευνα λοιπόν στη Διδακτική της Πληροφορικής περιλαμβάνει τα εξής στάδια:

1. Διατύπωση του συγκεκριμένου ερωτήματος. Προσδιορισμός θεωρητικών εργαλείων.
2. Επιστημολογική διερεύνηση: διερευνάται η "γένεση" των εννοιών που εμπλέκονται, τα προβλήματα που τις δημιούργησαν. Διερευνώνται επίσης οι σχέσεις των εννοιών αυτών με άλλες παρεμφερείς.
3. Επισκόπηση παρόμοιων εργασιών.
4. Προσδιορισμό των αντιλήψεων των σπουδαστών.
6. Κατασκευή διδακτικών καταστάσεων οι οποίες έχουν σα σκοπό τον μετασχηματισμό των αντιλήψεων των σπουδαστών.
7. Αξιολόγηση των διδακτικών καταστάσεων και βελτίωση τους.

Σύνοψη

Τα παραπάνω παραδείγματα φαίνονται να δείχνουν ότι σφάλματα και λάθη που δεν έχουν τυχαίο χαρακτήρα υφίστανται και στην κοινότητα των επαγγελματιών και των επιστημόνων προγραμματιστών. Στα ειδικευμένα τεχνικά και επιστημονικά περιοδικά μπορεί κανείς εύκολα να βρει παραδείγματα άρθρα και μελέτες για τα λανθασμένα προγράμματα (για παράδειγμα στο Gerhard S., Yelowitz L. [1976]) - και μάλιστα σε μεγάλο βαθμό. Ο J. Arsac [1983] θεωρεί ότι τα λάθη στον προγραμματισμό είναι τελικά τόσο συνηθισμένα ώστε η δημοσίευση λανθασμένων αλγορίθμων σε έγκυρα επιστημονικά περιοδικά δεν αποτελεί σκάνδαλο όπως θ' αποτελούσε η δημοσίευση λανθασμένων αποτελεσμάτων σε μαθηματικά περιοδικά. Δεν είναι βέβαιο ότι η δημοσίευση λανθασμένων αποτελεσμάτων σε μαθηματικά

περιοδικά αποτελεί σκάνδαλο (για σχετική συζήτηση δες και Hanna G. [1983], de Millo R.A. and als [1979] και Gasser J. [1989]) αλλά σίγουρα η δημοσίευση αλγορίθμων που περιέχουν λάθη δεν αποτελεί σκάνδαλο.

Το λάθος υφίσταται στην κοινότητα των προγραμματιστών κι επιπλέον είναι κι αναγνωρισμένο, νόμιμο κι αποτελεί κι αντικείμενο ερευνών. Οι πληροφορικοί εξάλλου κάνουν και διακρίσεις ανάμεσα στις έννοιες του λάθους, του σφάλματος, της εξαιρέσεως (Cristian F. [1985]). Παρ' όλ' αυτά υπάρχει ακόμη ένα ερώτημα: τα λάθη αυτά έχουν συστηματικό χαρακτήρα; Λάθη, για παράδειγμα, γίνονται σ' όλες τις ανθρώπινες δραστηριότητες κι είναι απίθανο να υπάρχουν μηχανικοί οι οποίοι δεν έχουν κάνει υπολογιστικά λάθη κατά τη σχεδίαση νέων συστημάτων. Αν θεωρήσουμε ότι η κατασκευή λογισμικού εντάσσεται στην ίδια κατηγορία ανθρώπινων δραστηριοτήτων με τη σχεδίαση μίας γέφυρας ή ενός πλοίου είναι φυσικό να υπάρχει ένα ποσοστό λαθών σε κάθε πρόγραμμα. Η θέση μας που υποστηρίζουμε είναι ότι ορισμένα από τα λάθη αυτά έχουν συστηματικό χαρακτήρα. Τα επιχειρήματα μας είναι δύο:

- Αρχικά το ποσοστό των λαθών είναι υπερβολικά υψηλό. Χωρίς να υπεισέλθουμε σε λεπτομέρειες, αναφέρουμε ότι σε ορισμένες ακραίες περιπτώσεις το κόστος εκσφαλμάτωσης, βελτίωσης κλπ είναι υπερπολλαπλάσιο του κόστους κατασκευής (αναφέρεται ενδεικτικά η περίπτωση λογισμικού με κόστος κατασκευής 75 \$/εντολή και κόστος συντήρησης κλπ 4000 \$/εντολή Boehm W. [1979] αλλά και Arzac J. [1983]). Ακόμη και σήμερα, το κόστος της βελτίωσης του λογισμικού είναι πολύ υψηλό και η δυσπιστία με την οποία οι χρήστες αντιμετωπίζουν τις πρώτες εκδόσεις νέων προγραμμάτων είναι χαρακτηριστική.

- Όμως όχι μόνο τα λάθη του λογισμικού είναι πολυπληθή αλλά συχνά κι οφείλονται σε λανθασμένη μεθοδολογία προγραμματισμού - αυτή είναι η θέση πολλών από τους επιφανέστερους επιστήμονες της Πληροφορικής (μερικά παραδείγματα θα δοθούν παρακάτω). Αυτό υπήρξε και το βασικό κίνητρο για τη δημιουργία στα τέλη της δεκαετίας του 1960 της «σχολής» του δομημένου προγραμματισμού. Η χρήση λανθασμένης μεθοδολογίας - ή μάλλον η έλλειψη μεθόδου - οδηγεί νομοτελειακά σε λάθη συχνά, λάθη συστηματικά, χρονοβόρα και πολλές φορές καταστροφικά, λάθη, εν γένει, με υψηλό οικονομικό και κοινωνικό κόστος.

Οι σπουδαστές

Όπως θα προσπαθήσουμε να δείξουμε, πολύ συχνά οι σπουδαστές κατασκευάζουν τα προγράμματα τους βασιζόμενοι απλά σε μια «νοητή εκτέλεση» (mental execution) ή μια «νοητή προσομοίωση» (mental simulation) ακολουθώντας έτσι την ίδια τεχνική την οποία

ακολουθούσαν οι προγραμματιστές αυθόρμητα κατά την πρώτη περίοδο της ανάπτυξης της πληροφορικής και η οποία ακριβώς καταπολεμήθηκε από τη σχολή του δομημένου προγραμματισμού και τις μοντέρνες μεθοδολογίες κατασκευής κι επικύρωσης λογισμικού - ήταν δηλαδή ένα από τα βασικά αίτια γι' αυτήν τη σχεδόν «αλλαγή παραδείγματος». Έτσι η μελέτη της «προγραμματιστικής» συμπεριφοράς των σπουδαστών που προτείνουμε φαίνεται να είναι δικαιολογημένη και σημαντική. Κεντρικό άξονα της εργασίας μας αποτελεί λοιπόν και η δημιουργία κι ανάλυση διδακτικών καταστάσεων οι οποίες αποσκοπούν ακριβώς στον μετασχηματισμό των λανθασμένων αντιλήψεων των σπουδαστών

Μερικές πρόσθετες παρατηρήσεις

Η ύπαρξη αυτών των συγκεκριμένων δυσκολιών που αντιμετωπίζουν οι σπουδαστές και τις οποίες ονομάσαμε με το γενικό όνομα "δυσλειτουργίες", παραμένει σε μεγάλο βαθμό μια ιδιαίτερη διαπίστωση όσων ακριβώς διδάσκουν προγραμματισμό. Κατά παράδοξο (αλλά όχι ανεξήγητο) τρόπο η διαδεδομένη αντίληψη είναι η αντίθετη. Το γεγονός δηλαδή ότι η διδασκαλία της Πληροφορικής, ή ακριβέστερα του προγραμματισμού, έχει πολύ νεαρή ηλικία, όπως εξάλλου κι η ίδια η Πληροφορική, δυσκολεύει, κατά κάποιο τρόπο την *κοινωνική αναγνώριση* των δυσκολιών αυτών: για παράδειγμα η ύπαρξη δυσκολιών στη διδασκαλία - ή μάλλον στην εκμάθηση - των Μαθηματικών ή των Φυσικών επιστημών αναγνωρίζεται ακόμη και θεσμικά. Το γεγονός ότι «τα Μαθηματικά είναι δύσκολα» όχι μόνο θεωρείται κοινός τόπος, αλλ' επιπλέον στις περισσότερες χώρες ο κλάδος της Διδακτικής των Μαθηματικών αποτελεί έναν κλάδο ερευνών επιστημονικά και θεσμικά κατοχυρωμένο. Δεν συμβαίνει όμως ακόμα αυτό με τον προγραμματισμό - τουλάχιστον όχι παντού. Επιπρόσθετα, η διδασκαλία του προγραμματισμού, λόγω της ιδιαίτερης ανάπτυξης της ίδιας της Πληροφορικής και της σχέσης της με τις εμπορικές και τεχνικές εφαρμογές, έχει συχνά έναν διττό χαρακτήρα: αποτελεί ένα αντικείμενο διδασκαλίας ισότιμο με τα Μαθηματικά, την Πολιτική Οικονομία ή τη Χημεία, το οποίο διδάσκεται στη δευτεροβάθμια ή την τριτοβάθμια εκπαίδευση με κοινό, στόχους και διδακτέα ύλη σαφώς καθορισμένα, αλλά αποτελεί συχνότατα κι αντικείμενο διδασκαλίας πολλών σχολών επιμόρφωσης, επαγγελματικής εκπαίδευσης και κατάρτισης, αντικείμενο απευθυνόμενο σε κοινό πολυποίκιλο : πτυχιούχους Πανεπιστημίου που επιμορφώνονται στη «χρήση των Η.Υ.», σε μελλοντικούς εκπαιδευτικούς οι οποίοι με τη σειρά τους θα το διδάξουν σε μαθητές, σε απόφοιτους Λυκείου οι οποίοι αποζητούν ένα επαγγελματικό εφόδιο, σε διάφορα ταχύρυθμα ή μη σεμινάρια που διοργανώνουν ιδιωτικοί και δημόσιοι φορείς κλπ. Η *διεσπαρμένη* αυτή διδασκαλία του προγραμματισμού αναστέλλει, κατά κάποιο τρόπο τη

συγκέντρωση στοιχείων και την οργανωμένη, συλλογική αντιμετώπιση των δυσκολιών αυτών.

Ίσως ο ιδιαίτερος χαρακτήρας *παιχνιδιού* που πήρε ο προγραμματισμός - μάλλον η Πληροφορική - και της *κοινωνικής έκτασης* του, να συντελεί επίσης στην μη αναγνώριση των ιδιαίτερων δυσκολιών της διδασκαλίας του. Για παράδειγμα πριν από μερικά χρόνια στο γαλλικό περιοδικό *Esprit* (τεύχος Φεβρουαρίου 1985) δύο υπεύθυνοι *animateurs* του *Club Méditerranée* (ενός είδους τουριστικού οργανισμού) εξηγούσαν πώς μάθαιναν στους συμμετέχοντες στα τουριστικά προγράμματα του *Club* τον προγραμματισμό και την «απλούστατη» BASIC σε 10 ώρες κατά τη διάρκεια των διακοπών, δίπλα στη θάλασσα (αναφέρεται από τον *Arsac J.* [1987], σελ. 24). Για να γίνει αντιληπτός αυτός ο χαρακτήρας *παιχνιδιού* στον οποίο αναφερθήκαμε, αρκεί ν' αντικαταστήσει κανείς τον «προγραμματισμό» με την «Φυσική»: όχι μόνο κανείς δεν θα διανοείτο να την διδάξει στη διάρκεια διακοπών στο *Club Méditerranée*, αλλά και μόνος ο ισχυρισμός ότι μαθαίνει κανείς τις βάσεις της, άνετα, σε 10 ώρες είναι αυτόχρημα γελοίος. Στην πραγματικότητα το να διδάξει κανείς Φυσική στην ακρογιαλιά δεν θα αποτελούσε καν διαφημιστικό επιχείρημα - μπορούμε με σιγουριά να ισχυριστούμε ότι μάλλον θ' απωθούσε τους υποψήφιους πελάτες παρά θα τους προσείλκυε.

Στην ίδια λογική του *παιχνιδιού* εντάσσεται επίσης το γεγονός ότι συχνά οι προσωπικοί Η.Υ. υπολογιστές βρίσκονται στα μεγάλα καταστήματα στο τμήμα των παιχνιδιών και η εμπορική τους κίνηση αυξάνει κατακόρυφα τα Χριστούγεννα. Ενώ όμως αναγνωρίζεται ότι τα "πειράματα Φυσικής" που πουλιούνται στα καταστήματα δεν έχουν παρά μια πολύ μακρινή σχέση με τη Φυσική των επαγγελματιών κι ερευνητών Φυσικών, αντίθετα η αγορά ενός προσωπικού υπολογιστή έχει πολύ συχνά ένα χαρακτήρα "επένδυσης". Αυτή η εικόνα του προγραμματισμού είναι σε πλήρη αντίθεση με τη θέση - για παράδειγμα - του *Dijkstra E. W.* [1982b], για τον οποίο :

Programming is one of the most difficult branches of applied mathematics; the poorer mathematicians had better remain pure mathematicians

Η *δημόσια εικόνα* λοιπόν ενός προγραμματισμού που μπορεί να γίνει αντιληπτός γρήγορα κι άνετα σε λίγες ώρες ακόμη και στις διακοπές κι αποτελεί και *παιχνίδι*, αυτός ο προγραμματισμός-παιχνίδι όχι μόνο είναι ασυμβίβαστος με έναν προγραμματισμό που «... αποτελεί έναν από τους δυσκολότερους κλάδους των εφαρμοσμένων μαθηματικών...» αλλά και με την συνακόλουθη ιδέα των δυσκολιών που παρουσιάζονται κατά την διδασκαλία και την εκμάθηση του.

Βέβαια ο *Dijkstra E. W.* και το *Club Mediterranee* δεν αναφέρονται στον ίδιο προγραμματισμό

- όπως δεν είναι ίδια τα Μαθηματικά της προσχολικής ηλικίας και των ερευνητών μαθηματικών. Ότι όμως εννοείται με το γενικό όρο «προγραμματισμός» συχνά θεωρείται ως μια διανοητική δραστηριότητα η οποία όμως είναι πολύ απλή ή είναι ισοδύναμη με την εκμάθηση ορισμένων τεχνικών στοιχείων (για παράδειγμα των χαρακτηριστικών ενός Η.Υ. και μίας γλώσσας προγραμματισμού).

Είναι για παράδειγμα πασίγνωστο ότι για δεκαετίες - κι ίσως ακόμη και σήμερα - βιβλία που φέρουν στον τίτλο τους την έκφραση "προγραμματισμός Η.Υ." στην πραγματικότητα αποτελούν οδηγούς χρήσης μιας γλώσσας προγραμματισμού - ενώ μέχρι και πρόσφατα τα σχολικά εγχειρίδια στο μάθημα της Πληροφορικής είχαν ως περιεχόμενο έναν οδηγό χρήσης των windows (!).

Οι παλαιότερες εργασίες

Μια συνοπτική παρουσίαση μερικών σημαντικών ευρημάτων της διδακτικής της Πληροφορικής είναι χρήσιμη γιατί με μια γενική έννοια τα αποτελέσματα των σχετικών ερευνών "συνοψίζουν", κατά κάποιο τρόπο, αποτελούν ένα είδος, περιληπτικής παρουσίασης των αποτελεσμάτων της διδακτικής κι άρα δίνουν μια γενική εικόνα, χρήσιμη για να αντιληφθεί ο αναγνώστης το σημερινό επίπεδο γνώσεων μας στη διδακτική της Πληροφορικής.

Οι πρώτες εργασίες για την διδασκαλία και την εκμάθηση του προγραμματισμού εμφανίζονται στα τέλη της δεκαετίας του 1960, με προεξάρχον το κλασσικό πλέον βιβλίο του G. M. Weinberg για την *ψυχολογία του προγραμματισμού* (Weinberg G. M. [1969]). Βασικός άξονας του βιβλίου είναι η προσπάθεια για βελτίωση της προγραμματιστικής δραστηριότητας (δηλαδή βελτίωση της απόδοσης) σε ατομικό και συλλογικό επίπεδο. Ο Weinberg επισημαίνει ήδη μερικά σημεία τα οποία στα επόμενα χρόνια βρέθηκαν στο κέντρο της διαμάχης γύρω από το δομημένο προγραμματισμό. Σημειώνει, για παράδειγμα, το γεγονός ότι ο προγραμματισμός αποτελεί βέβαια μια ατομική δραστηριότητα, αλλά αποτελεί και μια δραστηριότητα συλλογική κι έτσι τα παραγόμενα προγράμματα προορίζονται πολλές φορές να χρησιμοποιηθούν από ανθρώπους - ο οποίοι θα τα ελέγξουν, θα τα διορθώσουν, θα τα βελτιώσουν, θα τα μεταφέρουν σε άλλο περιβάλλον κλπ. Κατά συνέπεια τα προγράμματα πρέπει, κατά κάποιο τρόπο, να είναι "αναγνώσιμα" - να είναι δηλαδή γραμμένα κατά τρόπο τέτοιο ώστε να "απευθύνονται" και σε ανθρώπους. Είναι ίσως αξιοσημείωτο το γεγονός ότι σχεδόν 30 χρόνια μετά, οι τελευταίες εργασίες του D.E. Knuth, σχετικά με το περιβάλλον προγραμματισμού CWEB, έχουν τον ίδιο σκοπό - όπως ο ίδιος

αναφέρει σε συνέντευξη του¹. Ο Weinberg επισημαίνει επίσης το γεγονός ότι ο όρος "προγραμματισμός" καλύπτει δραστηριότητες οι οποίες συστεγάζονται υπό την ίδια επωνυμία, αλλά διαφέρουν ως προς το περιεχόμενό τους:

The high school student and the engineer represent two ends of a rich spectrum of programmers. These ends may or may not be different...But.... there is a difference

(ibid, σελ. 122)

Τούτο λοιπόν σημαίνει ότι τα προβλήματα στο ένα επίπεδο είναι διαφορετικά από τα προβλήματα στο άλλο επίπεδο και κατά συνέπεια κι οι λύσεις που μπορούν να προταθούν δεν μπορεί παρά να διαφέρουν.

Είναι επίσης αξιοσημείωτο ότι ο Weinberg αναφερόμενος στην εκμάθηση του προγραμματισμού θεωρεί ως κατάλληλες μεθόδους *την ενδοεπισκόπηση, την παρατήρηση γενικότερα, τον πειραματισμό*. Στο σημείο αυτό τάσσεται μάλλον με την πλευρά των πληροφορικών οι οποίοι θεωρούν ότι ο προγραμματισμός ως δραστηριότητα δεν ήταν ακόμη επιστήμη - στο τέλος της δεκαετίας του '60 - κι άρα δεν ήταν δυνατόν να διδαχθεί ως τέτοιος. Ο προγραμματισμός αποτελεί περισσότερο μια "τέχνη" (μια εμπειρική τέχνη) η οποία μεταδίδεται με όσμωση από τον "δάσκαλο" στο μαθητή αλλά δεν μπορεί να διδαχθεί με τις κλασσικές μεθόδους διδασκαλίας.

Το βιβλίο του G. M. Weinberg απετέλεσε την πρώτη συστηματική καταγραφή *του ρόλου του ανθρώπινου παράγοντα* στον προγραμματισμό. Οι συχνότερες αναφορές στο βιβλίο αυτό οδηγούν στο συμπέρασμα ότι το βιβλίο άσκησε μεγάλη επιρροή στους προγραμματιστικούς κύκλους.

Τα τέλη της δεκαετίας του '60 κι οι αρχές της δεκαετίας του '70 χαρακτηρίζονται από μια σειρά προσπαθειών της ακαδημαϊκής κοινότητας για την βελτίωση της εκπαίδευσης στην Πληροφορική, ακριβέστερα στον προγραμματισμό - αναφέρουμε ενδεικτικά της κατασκευής της PASCAL και της LOGO (όπως είναι γνωστό και οι δυο κατασκευάστηκαν αποκλειστικά για εκπαιδευτικούς σκοπούς), την οργάνωση συνεδρίων (όπως αυτό του [1972] στις *Sèvres*, Paris ή την *Working Conference on Programming Teaching Techniques*, IFIP TC-2 κά). Συνήθως όμως οι αναφορές της περιόδου αυτής δεν στηρίζονται σε θεωρητικές θέσεις ή πειραματικά δεδομένα αλλά συχνότερα αποτελούν απόψεις βασισμένες σε προσωπικές εμπειρίες, πεποιθήσεις και ιδεολογικές θέσεις. Οι αναφορές πραγματοποιήθηκαν κυρίως από εκπροσώπους της σχολής του δομημένου προγραμματισμού - οι οποίοι θεωρούν ως ένα ισχυρό κίνητρο για την καθιέρωση του δομημένου προγραμματισμού τη σαφή του υπεροχή

¹ συνέντευξη που εντοπίζεται εύκολα στο Internet.

έναντι των άλλων μεθόδων και στο πεδίο της διδασκαλίας και συνακόλουθα της εκμάθησης του προγραμματισμού. Λεπτομερέστερες αναφορές σε αυτές θα γίνουν σε διάφορα σημεία του κεφαλαίου 2.

Πιο σύγχρονες εργασίες

Στην δεκαετία του 1980 και του 1990 - αλλά και κατά την τρέχουσα δεκαετία - υπάρχει μια άνθηση ερευνών και γενικότερα εργασιών γύρω από θέματα διδασκαλίας κι εκμάθησης του προγραμματισμού. Η έξαρση αυτή έμμεσα σημαίνει την αναγνώριση της αναγκαιότητας για διερεύνηση των σχετικών θεμάτων, σημαίνει δηλαδή την αναγνώριση του γεγονότος ότι η διδασκαλία της Πληροφορικής (του προγραμματισμού ακριβέστερα) φαίνεται να είναι λιγότερο αποτελεσματική απ' ότι θα έπρεπε κι άρα πρέπει να βελτιωθεί. Αυτή η τάση διαγιγνώσκεται κι από γενικότερους δείκτες:

- την αυξημένη παγκοσμίως σχετική συνεδριακή δραστηριότητα: ανά τον κόσμο διοργανώνονται πολλά συνέδρια, συμπόσια κλπ με θέματα σχετικά με τη διδακτική και τη διδασκαλία της Πληροφορικής.
- την εμφάνιση σχετικών άρθρων στα μεγαλύτερα περιοδικά του είδους (όπως CACM κλπ). Είναι αξιοσημείωτο το γεγονός ότι αναφορές στα σχετικά θέματα έχουν κάνει - και μάλιστα με μεγάλη συχνότητα - μερικοί από τους πλέον γνωστούς παγκοσμίως επιστήμονες Πληροφορικής (Dijkstra, Knuth, Papert,...), ενώ περιοδικά όπως το CACM καθιέρωσαν πλέον μόνιμη στήλη για το θέμα της εκπαίδευσης.
- την ύπαρξη σχετικών ερευνητικών ομάδων και ειδικών περιοδικών (SIGSE και SIGUCE της ACM για παράδειγμα) και τη δημιουργία ειδικών επιτροπών για την εκπόνηση αναλυτικών προγραμμάτων για τη διδασκαλία της Πληροφορικής.
- τη δημιουργία μεγάλου όγκου εκπαιδευτικού λογισμικού (του λεγόμενου "διδασκτικού") για τη διδασκαλία θεμάτων Πληροφορικής και ειδικότερα του προγραμματισμού (χαρακτηριστικά αναφέρεται ότι υπάρχει κι ελληνική παραγωγή εκπαιδευτικών προγραμμάτων σχετικών με την πληροφορική και μάλιστα το πρώτο απ' αυτά - ένα "πολυμεσικό" πρόγραμμα παραγωγής ουσιαστικά του ΕΜΠ - αναφέρεται στη διδασκαλία του προγραμματισμού).

Ο μεγάλος αυτός όγκος δραστηριοτήτων πάσης φύσεως που σχετίζονται με τη διδασκαλία/εκμάθηση του προγραμματισμού αποτελεί έναν αδιαμφισβήτητο δείκτη της σπουδαιότητας που απέκτησε σταδιακά η διερεύνηση των σχετικών θεμάτων.

Αν και κατά την τελευταία εικοσαετία πραγματοποιήθηκαν πολλές έρευνες γύρω από τη διδασκαλία και την εκμάθηση του προγραμματισμού, ωστόσο ο κύριος όγκος των

ερευνητικών εργασιών περιστράφηκε γύρω από ορισμένα θέματα τα οποία εξετάστηκαν από πολλές οπτικές γωνίες.

Ένα πολύ μεγάλο μέρος των σχετικών ερευνών επικεντρώθηκε στη γλώσσα LOGO. Η γλώσσα LOGO κι η διδακτική της χρήση, αποτελούν ένα ιδιαίτερο κεφάλαιο το οποίο δεν θα εξετάσουμε εδώ. Υπενθυμίζεται ότι η γλώσσα LOGO αποτελεί ένα υποσύνολο της LISP με δυνατότητες διαχείρισης μιας "χελώνας" που σχεδιάζει στην οθόνη (ή στο έδαφος) κι άρα ως γλώσσα προγραμματισμού εντάσσεται στην κατηγορία των γλωσσών συναρτησιακού προγραμματισμού. Αν κι ο συναρτησιακός προγραμματισμός ως εισαγωγικό μάθημα στους αλγόριθμους και τον προγραμματισμό έχει πολλούς θιασώτες (για παράδειγμα το MIT και το Πανεπιστήμιο της Κύπρου χρησιμοποιούν τη γλώσσα Scheme, το Πανεπιστήμιο της Grenoble σε ορισμένες περιπτώσεις τη LOGO και σε άλλες τη LISP σε εισαγωγικά μαθήματα προγραμματισμού - δεξ αντίστοιχα Abelson H. [1985], Keravnou E.T. [1995], Viviet G. [1988]), ωστόσο η παρούσα εργασία ερευνά τα σχετικά με τον κλασικό "επιτακτικό" προγραμματισμό³ κι έτσι δεν θα εξεταστούν οι εργασίες οι σχετικές με τη LOGO .

Εκτός από τη LOGO, τα υπόλοιπα αντικείμενα τα οποία διερευνήθηκαν υπήρξαν σχετικά περιορισμένα: ιδιαιτερότητες του προγραμματισμού σε σχέση με άλλα αντικείμενα (π.χ. μαθηματικά, φυσική κλπ), θέματα σχετικά με μερικά βασικά δομικά στοιχεία των προγραμμάτων όπως οι μεταβλητές, οι επαναληπτικές δομές (οι βρόχοι), οι δομές επιλογών (iff...then...else), η διαχείριση των συμβολοσειρών (strings). Βέβαια υπήρξαν κι ερευνητικές εργασίες σε άλλα είδη θεμάτων όπως για παράδειγμα τη διδασκαλία του λογικού προγραμματισμού και τις σχετικές αντιλήψεις των σπουδαστών - ο αριθμός όμως των εργασιών αυτών υπήρξε περιορισμένος (δες Hook K. et als [1988], Taylor J. et als [1989]).

5. ΟΙ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟΙ ΜΙΚΡΟΚΟΣΜΟΙ: ΜΙΑ ΓΕΝΙΚΗ ΘΕΩΡΗΣΗ

Με τον όρο προγραμματιστικά περιβάλλοντα εννοούμε συστήματα τα οποία επιτρέπουν την έκδοση και εκτέλεση προγραμμάτων. Στην παρούσα θα ασχοληθούμε μόνο με μια ιδιαίτερη κατηγορία προγραμματιστικών περιβαλλόντων, τους προγραμματιστικούς *μικρόκοσμους*. Οι μικρόκοσμοι αυτοί αποτελούν μια ιδιαίτερη κατηγορία προγραμματιστικών περιβαλλόντων, αφού παρουσιάζουν μια σειρά κοινών χαρακτηριστικών. Περιπτώσεις που υπάγονται στην κατηγορία αυτή αποτελούν και η υλοποίηση της μηχανής του Post, αλλά και άλλων μικρόκοσμων όπως και παλιότερα διαδεδομένοι και οι χάρτινοι micro-assemblers.

³ Procedural programming, σε αντιδιαστολή με τον συναρτησιακό προγραμματισμό - functional programming- και τον λογικό - logic programming.

Ανακεφαλαιώνοντας συμπεράσματα από προηγούμενες ενότητες, η διδασκαλία και εκμάθηση του προγραμματισμού, όπως είναι γνωστό, παρουσιάζει αρκετές δυσκολίες. Τις τελευταίες δεκαετίες, έχει γίνει αξιόλογη ερευνητική δουλειά στην περιοχή της *Διδακτικής της Πληροφορικής* και ειδικότερα στην περιοχή της διδασκαλίας του προγραμματισμού. Όπως προκύπτει από τη σχετική βιβλιογραφία (Δαγδiléλης 1996a), ένας από τους σημαντικότερους παράγοντες που έχει διαπιστωθεί ότι αποτελεί πηγή δυσκολιών για την εκμάθηση του προγραμματισμού έγκειται στο γεγονός ότι, όπως φαίνεται, η *κλασική προσέγγιση διδασκαλίας* (Brusilovsky et als 1997) είναι ασύμβατη με τις πραγματικές διδακτικές ανάγκες των σπουδαστών. Με τον όρο κλασική προσέγγιση διδασκαλίας, όπως αναφέρθηκε και σε προηγούμενη παράγραφο, εννοούμε τη διδασκαλία που συνίσταται:

- στη χρήση μιας γλώσσας γενικού σκοπού (όπως οι Pascal, C, κλπ),
- ενός επαγγελματικού περιβάλλοντος προγραμματισμού για τη γλώσσα αυτή, και
- στην επίλυση ενός συνόλου προβλημάτων επεξεργασίας αριθμών και συμβόλων.

Στα επόμενα, παρουσιάζονται αναλυτικά τα χαρακτηριστικά της κλασικής προσέγγισης διδασκαλίας του προγραμματισμού και των κυριότερων *διδακτικών προβλημάτων* που εμφανίζονται στο πλαίσιο της. Στη συνέχεια δίνεται μια σύντομη περιγραφή των *εναλλακτικών προσεγγίσεων διδασκαλίας του προγραμματισμού* που αναπτύχθηκαν έχοντας ως στόχο τη στήριξη της διδακτικής πράξης και την εξάλειψη των σχετικών διδακτικών προβλημάτων.

Όπως έχει προαναφερθεί, σύμφωνα με αρκετούς ερευνητές, η υιοθέτηση της κλασικής προσέγγισης διδασκαλίας του προγραμματισμού αποτελεί έναν από τους σημαντικότερους παράγοντες που καθιστά την εκμάθηση του προγραμματισμού δύσκολη. Τα σημαντικότερα προβλήματα που αντιμετωπίζουν οι αρχάριοι προγραμματιστές που διδάσκονται τις αρχές του προγραμματισμού με την κλασική προσέγγιση, όπως προκύπτει από εμπειρικές παρατηρήσεις και μελέτες ερευνητών αλλά και μαρτυρίες των ίδιων των σπουδαστών, είναι τα εξής:

- *Οι γλώσσες προγραμματισμού γενικού σκοπού διαθέτουν, κατά κανόνα, ένα μεγάλο ρεπερτόριο εντολών και είναι πολύπλοκες.* Το υλικό που πρέπει να καλύψει ο σπουδαστής, το οποίο περιλαμβάνει το θεωρητικό πλαίσιο μιας γλώσσας προγραμματισμού σε συνδυασμό με τις βασικές αρχές του προγραμματισμού, είναι πολύ μεγάλο. Το γεγονός αυτό έχει ως αποτέλεσμα ο σπουδαστής να δυσκολεύεται να

κατανοήσει όλο το υλικό και να αποκτήσει το απαιτούμενο υπόβαθρο. (Brusilovsky et als 1997, Ruckert & Halpern 1993)

- Η προσοχή των μαθητών επικεντρώνεται στην εκμάθηση της σύνταξης της γλώσσας και όχι στην ανάπτυξη ικανοτήτων επίλυσης προβλημάτων (Brusilovsky et als 1997; Freund & Roberts 1996; Studer et als 1995; Ziegler & Crews 1999). Οι συντακτικοί κανόνες των γλωσσών προγραμματισμού γενικού σκοπού είναι πολύπλοκοι, με αποτέλεσμα οι σπουδαστές να αφιερώνουν πολύ χρόνο στην εκμάθηση της σύνταξης της γλώσσας προγραμματισμού. Ενώ λοιπόν ο σκοπός ενός μαθήματος εισαγωγής στον προγραμματισμό είναι η χρήση μιας γλώσσας προγραμματισμού για τη διδασκαλία εννοιών του προγραμματισμού, την υλοποίηση αλγορίθμων και την ανάπτυξη ικανοτήτων επίλυσης προβλημάτων, τελικά η προσοχή επικεντρώνεται στη γλώσσα προγραμματισμού.
- Το προγραμματιστικό περιβάλλον συνήθως δεν παρέχει δυνατότητες οπτικοποίησης (η διαδικασία εκτέλεσης ενός προγράμματος δεν είναι ορατή από τον σπουδαστή) και επομένως *δεν υπάρχει, στήριξη του σπουδαστή στην κατανόηση των βασικών ενεργειών και δομών ελέγχου*. Εφόσον ο σπουδαστής δεν μπορεί να ακολουθήσει οπτικά την εκτέλεση ενός προγράμματος, το αντιμετωπίζει απλά ως μια διαδικασία εισόδου-εξόδου, ή, γεγονός που μπορεί να αποδειχθεί διδακτικά πιο επιζήμιο, κάνει υποθέσεις για τον τρόπο λειτουργίας του συστήματος, βασισμένος στις εμπειρίες του. (Brusilovsky et als 1997, Ross 1991, Sangwan et als 1998)
- *Οι εμπορικοί μεταγλωττιστές δεν ικανοποιούν τις ανάγκες των αρχάριων προγραμματιστών*. Οι περισσότεροι εμπορικοί μεταγλωττιστές απευθύνονται σε επαγγελματίες και όχι σε σπουδαστές, προκαλώντας έτσι αρκετά προβλήματα όταν χρησιμοποιούνται στο πλαίσιο ενός μαθήματος εισαγωγής στον προγραμματισμό. (Freund & Roberts 1996, Schorsch 1995)
- *Η διανοητική πολυπλοκότητα που απαιτεί η εκφορά ενός αλγορίθμου σε μια γλώσσα προγραμματισμού είναι μεγάλη*. Οι προγραμματιστές πολύ συχνά κατά την ανάπτυξη προγραμμάτων δημιουργούν νοητά μοντέλα του προγράμματός τους, του τρόπου λειτουργίας του και χρήσης των δεδομένων. Η διανοητική «απόσταση» ανάμεσα στις νοητές αναπαραστάσεις των αλγορίθμων ενός αρχάριου προγραμματιστή και στην περιγραφή τους σε μια γλώσσα προγραμματισμού είναι πολύ μεγάλη, εξαιτίας της «φύσης» της γλώσσας. (Smith et als 1994; Pane & Myers 1996)
- Η επίλυση ενδιαφερόντων προβλημάτων απαιτεί την εκμάθηση ενός μεγάλου υποσυνόλου της γλώσσας και την ανάπτυξη αρκετά μεγάλων προγραμμάτων, απαιτεί δηλαδή την επικέντρωση της προσοχής στην εκμάθηση της γλώσσας (Brusilovsky et als 1997).

Το γεγονός ότι η κλασική προσέγγιση διδασκαλίας του προγραμματισμού δεν ικανοποιεί τις διδακτικές ανάγκες των σπουδαστών είχε ως αποτέλεσμα την ανάπτυξη ειδικών μεθοδολογιών διδασκαλίας, γλωσσών προγραμματισμού και εκπαιδευτικών εργαλείων. Στη συνέχεια περιγράφονται σύντομα οι βασικές προσεγγίσεις διδασκαλίας του προγραμματισμού που έχουν προταθεί από διακεκριμένους ερευνητές (Ξυνόγαλος 2002, Εφόπουλος 2005).

Μικρόκοσμοι – Μικρογλώσσες Προγραμματισμού

Η βασική ιδέα των *μικρόκοσμων* (microworlds) και των *μικρογλωσσών* (mini-languages) προγραμματισμού είναι η δημιουργία μιας γλώσσας προγραμματισμού που αποτελείται από ένα περιορισμένο ρεπερτόριο εντολών με απλή σύνταξη και σημασία. Οι βασικές ιδέες που διέπουν τη λειτουργία των μικροκόσμων είναι, έμμεσα, λύσεις στα προβλήματα που παρουσιάζουν τα δεδομένα περιβάλλοντα προγραμματισμού (τα οποία χρησιμοποιούνται κυρίως από ερευνητές ή επαγγελματίες).

Οικειότητα και ύπαρξη συγκεκριμένης οντότητας.

Ο σπουδαστής μαθαίνει τις βασικές έννοιες του προγραμματισμού ελέγχοντας ένα πρωταγωνιστή – μια χελώνα, ένα ρομπότ ή κάποια άλλη οντότητα – που ζει στον μικρόκοσμο (Κυνηγός και Στασινή, 202).

Τα προγραμματιστικά περιβάλλοντα που βασίζονται στην προσέγγιση των μικρόκοσμων ενσωματώνουν συνήθως δυναμικές προσομοιώσεις εκτέλεσης των προγραμμάτων (program animation). Ο σπουδαστής δηλαδή, έχει τη δυνατότητα να εκτελεί τα προγράμματα που αναπτύσσει βήμα προς βήμα, παρακολουθώντας το αποτέλεσμα εκτέλεσης της κάθε εντολής, στην κατάσταση του μικρόκοσμου. Βασικό πλεονέκτημα της προσέγγισης των μικρόκοσμων, εκτός βέβαια από το περιορισμένο ρεπερτόριο εντολών και την οπτικοποίηση της εκτέλεσης ενός προγράμματος, είναι και το γεγονός ότι οι μικρόκοσμοι βασίζονται – στην πλειοψηφία τους – σε κάποιο φυσικό μοντέλο, που είναι ήδη γνωστό στους σπουδαστές και όχι σε κάποιο άγνωστο και δυσνόητο (για τους σπουδαστές) μοντέλο αναφοράς, όπως η μηχανή του Von Neuman.

Οι προγραμματιστικοί μικρόκοσμοι χρησιμοποιούνται για την εισαγωγή των σπουδαστών στον προγραμματισμό, εδώ και αρκετά χρόνια. Ένας μεγάλος αριθμός διδασκόντων τόσο σε Πανεπιστήμια, όσο και σε σχολεία, από διαφορετικές χώρες, αναφέρουν θετικά αποτελέσματα από τη χρήση προγραμματιστικών μικρόκοσμων (Brusilovsky et als 1997). Ωστόσο, πρέπει να επισημανθεί ότι τα αποτελέσματα αυτά βασίζονται περισσότερο στις εντυπώσεις των διδασκόντων, παρά σε εμπειρική αξιολόγηση της αποτελεσματικότητας των

μικρόκοσμων. Επιπλέον, οι εργασίες που παρουσιάζουν περιπτώσεις χρήσης μικρόκοσμων στην τάξη ή ειδικά πειράματα με στόχο την αξιολόγηση τους ως διδακτικά εργαλεία είναι λίγες και στις περισσότερες περιπτώσεις δεν είναι επαναλαμβανόμενες, έτσι ώστε τα αποτελέσματα τους να θεωρούνται απολύτως έγκυρα (Ξυνόγαλος 2005).

Βέβαια, η αξιολόγηση της διδακτικής προσέγγισης που βασίζεται στη χρήση των μικρόκοσμων δεν είναι εύκολη και θα πρέπει σε κάθε περίπτωση να λαμβάνονται υπ' όψη οι ιδιαίτερες συνθήκες χρήσης τους, το προφίλ των σπουδαστών και τα ιδιαίτερα χαρακτηριστικά του προγραμματιστικού μικρόκοσμου που χρησιμοποιείται. Αν και αρκετοί ερευνητές επισημαίνουν ότι οι μικρόκοσμοι μπορούν να χρησιμοποιηθούν αποτελεσματικά σε κάθε περίπτωση δεν υπάρχουν επαρκή στοιχεία που να το επιβεβαιώνουν. Οι (Pattis et als 1995, Brusilovsky et als 1997) επισημαίνουν ότι οι μικρόκοσμοι μπορούν να χρησιμοποιηθούν:

- 1. στο πλαίσιο ενός μαθήματος εισαγωγής στον προγραμματισμό, το οποίο μπορεί να επικεντρωθεί αποκλειστικά στη διδασκαλία εννοιών του προγραμματισμού ή να προετοιμάσει παράλληλα τους σπουδαστές για την ευκολότερη μετάβαση σε μια γλώσσα προγραμματισμού γενικού σκοπού,*
- 2. σε προχωρημένα στάδια σπουδών για τη στήριξη των σπουδαστών στην κατανόηση δύσκολων εννοιών (π.χ. αναδρομή, δείκτες κ.τ.λ),*
- 3. για την ανάπτυξη δεξιοτήτων επίλυσης προβλημάτων και αλγοριθμικού τρόπου σκέψης,*
- 4. από άτομα που θέλουν να αποκτήσουν κάποιες βασικές γνώσεις γύρω από τον προγραμματισμό, αλλά και την επιστήμη των υπολογιστών γενικότερα.*

Επίσης, όπως ήδη αναφέρθηκε, στην αξιολόγηση της διδακτικής προσέγγισης των μικρόκοσμων, θα πρέπει οπωσδήποτε να λαμβάνονται υπ' όψη τα ιδιαίτερα χαρακτηριστικά του προγραμματιστικού περιβάλλοντος που χρησιμοποιείται, αφού ακόμα και για τον ίδιο μικρόκοσμο έχουν – σε αρκετές περιπτώσεις – υλοποιηθεί περιβάλλοντα με διαφορετικά χαρακτηριστικά και λειτουργίες. Αν και οι προγραμματιστικοί μικρόκοσμοι που έχουν αναπτυχθεί είναι πολλοί και με σημαντικές διαφορές, μπορούν να ταξινομηθούν σε γενικές κατηγορίες όπως οι μικρόκοσμοι βασισμένοι στη Logo, μικρόκοσμοι βασισμένοι στα ρομπότ Karel (Pattis et als 1995) και Karel++ (Bergin et als 1997), μικρόκοσμοι βασισμένοι σε φυσικά αντικείμενα, μικρόκοσμοι – παιχνίδια κ.ά.

Τα χαρακτηριστικά που ενσωματώνει η πλειοψηφία των μικρόκοσμων είναι: μινιμαλιστικό

και εύχρηστο γραφικό ενδιάμεσο (GUI), βήμα-προς-βήμα εκτέλεση, και τεχνικές οπτικοποίησης λογισμικού. Ορισμένοι μικρόκοσμοι ενσωματώνουν και κάποια άλλα χαρακτηριστικά, που παρέχουν επιπλέον στήριξη στους σπουδαστές, όπως: εκδότες δομής για την ευκολότερη ανάπτυξη προγραμμάτων και την αποφυγή συντακτικών λαθών, επεξηγηματική οπτικοποίηση, δυνατότητα αυτόματης δημιουργίας διαγραμμάτων ροής και ενσωματωμένες στο περιβάλλον δραστηριότητες (Ξυνόγαλος 2005, Εφόπουλος 2005).

Προσομιώσεις φυσικών αντικειμένων που λειτουργούν ως μικρόκοσμοι

Η χρήση ενός πραγματικού συστήματος του κόσμου ως μοντέλου αναφοράς καθιστά τους μικρόκοσμους κατάλληλους για τη διδασκαλία του προγραμματισμού ακόμα και σε μαθητές μικρής ηλικίας. Για την περαιτέρω στήριξη των μαθητών μικρής ηλικίας – ακόμα και προσχολικής – κατά την εισαγωγή τους στις βασικές έννοιες του προγραμματισμού, μπορούν να χρησιμοποιηθούν μικρόκοσμοι στους οποίους οι εντολές αποτελούν φυσικά αντικείμενα. Χαρακτηριστικά παραδείγματα αποτελούν η Slot machine που αναπτύχθηκε στα εργαστήρια του MIT και η AlgoBlock που αναπτύχθηκε από την ερευνητική ομάδα της NEC Corporation. Τα χαρακτηριστικά των συστημάτων αυτών συνοψίζονται στον ακόλουθο πίνακα (Ξυνόγαλος 2002, Ξυνόγαλος 2004, Ξυνόγαλος 2005):

Μικρόκοσμος	Ιδιαίτερα Χαρακτηριστικά
Slot machine (Kahn 1996)	Το ρεπερτόριο εντολών της είναι υποσύνολο της Logo και απευθύνεται σε παιδιά προσχολικής ηλικίας. Οι εντολές έχουν τη μορφή πλαστικών καρτών οι οποίες εισάγονται στις σχισμές μιας ειδικά σχεδιασμένης μηχανής. Κατά τη διάρκεια εκτέλεσης ενός προγράμματος φωτίζεται η σχισμή της οποίας εκτελείται η εντολή.
AlgoBlock (Suzuki & Kato 1995)	Για την ανάπτυξη των προγραμμάτων χρησιμοποιούνται κύβοι που συνδέονται μεταξύ τους. Απευθύνεται σε παιδιά μικρής ηλικίας. Το αποτέλεσμα της εκτέλεσης φαίνεται στην οθόνη με τη μορφή ενός υποβρυχίου που κινείται βάση των εντολών (κύβων), ενώ τη στιγμή που εκτελείται μια εντολή ανάβει ένα λαμπάκι στο πάνω μέρος του κύβου.

Παιχνίδια ως μικρόκοσμοι

Ένα από τα βασικά πλεονεκτήματα των μικρόκοσμων, έγκειται στο γεγονός ότι προκαλούν το

ενδιαφέρον των σπουδαστών. Σε ορισμένες περιπτώσεις μάλιστα, ένας μικρόκοσμος μπορεί στην ουσία να χαρακτηριστεί ως παιχνίδι. Χαρακτηριστικό παράδειγμα από την παλιότερη «γενιά» εκπαιδευτικών αποτελεί ο μικρόκοσμος AlgoArena (Kato & Ide 1995, Ξυνόγαλος 2004, Ξυνόγαλος 2005), ένα εκπαιδευτικό εργαλείο που αναπτύχθηκε στην Ιαπωνία για τη διδασκαλία της σχεδίασης λογισμικού σε μαθητές Γυμνασίου. Στην ουσία πρόκειται για ένα παιχνίδι που προσομοιώνει την πάλη Sumo, το παραδοσιακό σπορ της Ιαπωνίας. Οι μαθητές προγραμματίζουν τις ενέργειες του δικού τους παλαιστή χρησιμοποιώντας μια γλώσσα βασισμένη στη LOGO, προκειμένου να κερδίσουν κάποιον αντίπαλο (συνήθως κάποιον συμμαθητή ή καθηγητή). Το AlgoArena αποτελείται από το περιβάλλον προσομοίωσης της πάλης, ένα περιβάλλον γνωστό από τα ηλεκτρονικά παιχνίδια και τον εκδότη στον οποίο γράφονται τα προγράμματα. Ο εκδότης είναι έτσι σχεδιασμένος, ώστε οι μαθητές να μπορούν να αναπτύσσουν τα προγράμματά τους, επιλέγοντας σχεδόν όλες τις δεσμευμένες λέξεις με το ποντίκι από ένα μενού. Τα τελευταία χρόνια η κατεύθυνση αυτή (εκμάθηση μέσω παιχνιδιών) έχει αναπτυχθεί πολύ και θα την επανεξετάσουμε σε κατοπινή ενότητα.

Η «οικογένεια» περιβαλλόντων Logo (Logo-like environments)

Η γλώσσα LOGO είναι δημιούργημα του Αμερικανού ερευνητή Seymour Papert (1989, Papert 1991). Ο S. Papert υπήρξε μαθητής του J. Piaget και η γλώσσα LOGO υπήρξε καρπός της προσπάθειας του να κατασκευάσει έναν μικρόκοσμο στον Η.Υ., ο οποίος να «υλοποιεί» τη θεωρία του J. Piaget, τον κονστρουκτιβισμό (constructivism), αλλά και την επέκταση της θεωρίας αυτής που επινόησε ο ίδιος ο Papert, το λεγόμενο κονστρουκτιονισμό (constructionism).

Η βασική ιδέα της Logo, είναι η δημιουργία μιας «Μαθηματικοχώρας», δηλαδή ενός περιβάλλοντος που θα λειτουργεί με βάση τα Μαθηματικά. Ωστόσο, καθώς η χελώνα προγραμματίζεται, η γλώσσα Logo μπορεί να χρησιμοποιηθεί και στη διδασκαλία του προγραμματισμού – μάλιστα ο Brian Harvey από το Μ.Ι.Τ. έχει συγγράψει ένα τρίτομο έργο για προγραμματισμό με τη γλώσσα Logo (Harvey 1997a, Harvey 1997b, Harvey 1997c). Η γνωστή χελώνα της Logo εδώ και 40 σχεδόν χρόνια, έχει χρησιμοποιηθεί σε δεκάδες παραλλαγές, δημιουργώντας μια «οικογένεια» λογισμικών με ανάλογα χαρακτηριστικά. Τα χαρακτηριστικά της Logo είναι:

- Ένας «πρωταγωνιστής» που υποτίθεται ότι παριστάνει μια χελώνα και κινείται πάνω σε μια επίπεδη, λευκή επιφάνεια, χαράσσοντας μια γραμμή από τα σημεία από τα οποία περνάει.

- Μια απλή γλώσσα προγραμματισμού (συναρτησιακή, functional) για τον καθορισμό της κίνησης της χελώνας, με πολύ απλές δομές δεδομένων (λίστες), με στοιχειώδεις λειτουργίες εισόδου-εξόδου, με δυνατότητες αναδρομής και επεκτάσιμη, μέσω συναρτήσεων που δημιουργεί ο χρήστης.
- Επειδή η κίνηση της χελώνας είναι άμεση, ο χρήστης διαπιστώνει με προφανή τρόπο αν πέτυχε τον προκαθορισμένο στόχο του (για παράδειγμα αν η χελώνα «σχημάτισε» ένα τετράγωνο) και μπορεί να διορθώσει το πρόγραμμά του, σε περίπτωση λάθους.

Η οικογένεια της Logo περιλαμβάνει πολλές παραλλαγές του βασικού περιβάλλοντος. Επίσης, η γλώσσα Logo, ενσωματώνεται ως βασική γλώσσα προγραμματισμού σε διάφορα περιβάλλοντα (όπως το ABAKEIO, το MicroWorlds Pro, το HyperStudio κ.ά.). Ορισμένες παραλλαγές της Logo έχουν τρισδιάστατη αναπαράσταση, ή ενσωματώνουν διάφορες πρόσθετες ιδιότητες (για παράδειγμα, οι χελώνες έχουν «μάζα» και αδράνεια). Η οικογένεια Logo περιλαμβάνει και την «υπο-οικογένεια» Karel, ενός υποθετικού ρομπότ, προσομοιωμένου στην οθόνη, που κινείται πάνω σε ένα «πλέγμα» ορθογωνίως τεμνομένων δρόμων, με παραλλαγές όπως το Joseph, ο Karel++, ο objectKarel και άλλα, ανάλογα συστήματα. Μερικές αξιοσημείωτες περιπτώσεις συστημάτων της οικογενείας Logo είναι η LadyBug («πασχαλίτσα»), ένα πολύ απλό περιβάλλον στο οποίο ο προγραμματισμός πραγματοποιείται χωρίς καθόλου γραπτό κείμενο, η StarLogo, ένα περιβάλλον «μαζικά παράλληλου προγραμματισμού», στο οποίο ο χρήστης μπορεί να διαχειρίζεται ταυτόχρονα εκατοντάδες ή χιλιάδες «χελώνες» και τα Lego Mindstorms, ρομποτικές κατασκευές, εμπνευσμένες από τη γλώσσα Logo.

Με τα περιβάλλοντα αυτά ασχολούμαστε εκτενέστερα σε επόμενη ενότητα.

6. ΠΑΡΑΔΕΙΓΜΑΤΑ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΩΝ ΜΙΚΡΟΚΟΣΜΩΝ ΚΑΙ ΕΝΑΛΛΑΚΤΙΚΩΝ ΠΕΡΙΒΑΛΛΟΝΤΩΝ

Παρατίθενται διευθύνσεις για ορισμένους προγραμματιστικούς μικρόκοσμους οι οποίοι μπορούν να χρησιμοποιηθούν για εισαγωγικά μαθήματα (παρόμοιοι μικρόκοσμοι αναφέρονται και στην ενότητα για Logo-like περιβάλλοντα, αλλά και στο κεφάλαιο για τον αντικειμενοστραφή προγραμματισμό). Πολλά από τα περιβάλλοντα αυτά, τα λογισμικά και τις γλώσσες προγραμματισμού εντάσσονται σε μια λογική ενός "μονοπατιού μάθησης" (learning path), δηλαδή της διδασκαλίας μέσα σε περιβάλλοντα αυξανόμενης πολυπλοκότητας, έτσι ώστε στο τέλος οι μαθητές οι σπουδαστές να εξοικειωθούν με τα επαγγελματικά περιβάλλοντα. Το πιο χαρακτηριστικό παράδειγμα αποτελεί το βιβλίο των Η.

Ledgard & M. Marcotty (1987) The world of programming languages, στο οποίο οι σπουδαστές εισάγονται σταδιακά σε ολοένα και πιο σύνθετες έννοιες προγραμματισμού μέσα από μια σειρά mini-γλωσσών επινοημένων για διδακτικούς λόγους. Πολλά μικροπεριβάλλοντα περιγράφονται σε άλλες ενότητες (περιβάλλοντα για το δομημένο προγραμματισμό, Logo-like περιβάλλοντα, περιβάλλοντα για τον αντικειμενοστραφή προγραμματισμό, ρομποτικά περιβάλλοντα)

Περιβάλλοντα έκφρασης αλγορίθμων χωρίς Η.Υ.

Μπορεί να φαίνεται παράδοξη η διδασκαλία του προγραμματισμού χωρίς Η.Υ. αλλά συστήματα αυτού του είδους χρησιμοποιήθηκαν εκτεταμένα τις δεκαετίες του '70 και του '80, όταν δεν υπήρχαν μικροϋπολογιστές ή ήταν σχετικά ακριβοί και σπάνιοι. Ωστόσο, ακόμη και σήμερα ορισμένοι χρησιμοποιούνται για διδακτικούς σκοπούς. Σήμερα, το βασικό τους πλεονέκτημα παραμένει η εξαιρετική τους απλότητα, που τους κάνει να μοιάζουν περισσότερο ως παιχνίδια (με εξαίρεση μερικούς από αυτούς).

Το σύστημα **Little Man Computer (LMC)** επινοήθηκε περίπου το 1965 και ουσιαστικά στηρίζεται σε μια εξαιρετικά απλοποιημένη εκδοχή ενός Η.Υ. και μια γλώσσα προγραμματισμού assembly-like που περιλαμβάνει λιγότερες από 20 εντολές. Η λειτουργία του συστήματος μπορεί να προσομοιωθεί σε ένα χειροκίνητο (!), Η.Υ. τη λειτουργία του οποίου μπορεί κανείς να βρει στη wikipedia:

http://en.wikipedia.org/wiki/CARDboard_Illustrative_Aid_to_Computation

και μια επίδειξη στο YouTube:

<http://www.youtube.com/watch?v=CW96m7R0u-s>

Το Αμερικανικό αυτό σύστημα ονομάστηκε CARDIAC. Λεπτομέρειες για το σύστημα αυτό μπορούν να βρεθούν στο Διαδίκτυο και μια καλή πηγή αποτελεί η wikipedia: http://en.wikipedia.org/wiki/Little_Man_Computer. Οι δημιουργοί του CARDIAC, δημιούργησαν ένα πιο "ισχυρό" σύστημα με το όνομα SIMPLEX

(<http://www.mcmanis.com/chuck/computers/simplex.html>).

Κατά καιρούς έχουν δημιουργηθεί μικροπεριβάλλοντα που προσομοιώνουν τη λειτουργία του CARDIAC (και του Γαλλικού Ordinaroche - δες παρακάτω) καθώς και του SIMPLEX.

Τη δεκαετία του 1980, ένα Γαλλικό περιοδικά αφιέρωσε ένα τεύχος στο σύστημα αυτό (προσφέροντας δωρεάν ένα χειροκίνητο "Η.Υ." που ονομάστηκε στα Γαλλικά Ordinaroche, ένα πλήρες εγχειρίδιο οδηγιών και μερικά στοιχειώδη προγράμματα για το σύστημα LMC -

όπως ένα παιχνίδι "τριάρας" που παίζει ο "Η.Υ." εναντίον ανθρώπου).



Ανάλογο είναι και το σύστημα **Little Man 3** (LC-3, <http://en.wikipedia.org/wiki/LC-3>) όπως και το πιο σύγχρονο σύστημα **MicroSim** (<http://en.wikipedia.org/wiki/MikroSim>).

Πολλά από τα συστήματα αυτά (όπως το LC-3 και το MicroSim) επινοήθηκαν και χρησιμοποιήθηκαν (ή χρησιμοποιούνται ακόμη) για τη διδασκαλία μαθημάτων που έχουν να κάνουν με την Αρχιτεκτονική των επεξεργαστών κλπ.

Το 2012 παρουσιάστηκε μια απόπειρα για ένα νέο είδος παιχνιδιού - Η.Υ. το **Ox10^c** (<http://0x10c.com/>)

Οι υποθετικοί Η.Υ. MIX και MMIX

(<http://en.wikipedia.org/wiki/MIX>, <http://en.wikipedia.org/wiki/MMIX> καθώς και <http://www-cs-faculty.stanford.edu/~knuth/mmix.html>) επινοήθηκαν από τον D.E. Knuth και χρησιμοποιήθηκαν στα πολύ γνωστά βιβλία του *The Art of Computer Programming*. Οι φανταστικοί αυτοί υπολογιστές είναι σχετικά πολύπλοκοι στη λειτουργία τους.

Ενδιαφέρουσα είναι και η προσέγγιση του περιβάλλοντος **BALTIE-3** (<http://progopedia.com/language/baltie/>)

Ένας **χάρτινος Η. Υ.** που χρησιμοποιείται για μια εισαγωγή στον προγραμματισμό. Χρησιμοποιήθηκε επί πολλά χρόνια σε διδασκαλίες (ακόμη και Πανεπιστημιακού επιπέδου) και ουσιαστικά περιγράφει με συστηματικό τρόπο πρακτικές που ακολουθούν σχεδόν όλοι οι καθηγητές Πληροφορικής. Στην πραγματικότητα αυτός ο υποθετικός Η.Υ. αποτελεί μια μέθοδο για τη διδασκαλία του προγραμματισμού - ακριβέστερα αλγορίθμων. Η λεπτομερής περιγραφή του υπάρχει σε παράρτημα.

Περιβάλλοντα δημιουργίας παιχνιδιών

Το **Stagecast** <http://www.stagecast.com/> αποτελεί ένα τυπικό παράδειγμα περιβάλλοντος στο οποίο μπορούν να δημιουργηθούν διάφορα παιχνίδια και προσομοιώσεις (σύμφωνα με τους κατασκευαστές: με εύκολο τρόπο) ενώ στην ιστοσελίδα:

<http://www.stagecast.com/cgi-bin/templator.cgi?PAGE=School/LESSONWORLDS> υπάρχουν παραδείγματα προσομοιώσεων για την υποστήριξη της διδασκαλίας διαφόρων εννοιών, από ποικίλους επιστημονικούς κλάδους (Φυσική, μαθηματικά κ.ά.). Το περιβάλλον φαίνεται να είναι εξέλιξη ενός παλιότερου περιβάλλοντος που είχε αναπτυχθεί για τους H.Y. Macintosh της Apple, και σήμερα αποτελεί ένα περιβάλλον ανάπτυξης API, σύμφωνα με τη Wikipedia: http://en.wikipedia.org/wiki/Cocoa_%28API%29 (τελευταία επίσκεψη τον Οκτώβριο του 2012).

Το περιβάλλον **Alice** (έκδοση 3.1) αναπτύχθηκε στο Πανεπιστήμιο Carnegie-Mellon και είναι ένα περιβάλλον για προγραμματισμό που διατίθεται δωρεάν (<http://www.alice.org/>) ενώ υπάρχει και μια έκδοση storytelling:

(<http://www.alice.org/kelleher/storytelling/index.html>). Για τη δεύτερη αυτή έκδοση υπάρχει μια σχετική αρθρογραφία για τον τρόπο που χρησιμοποιείται στη διδασκαλία του προγραμματισμού:

(<http://www.alice.org/kelleher/storytelling/papers.html>).

(Τελευταία επίσκεψη τον Ιούλιο του 2012).

Kodu

Το **Kodu** είναι ένα πολύ ισχυρό περιβάλλον για τη διδασκαλία του προγραμματισμού μέσω της κατασκευής ψηφιακών παιχνιδιών. Διατίθεται δωρεάν από τη Microsoft.

Σε επόμενη ενότητα υπάρχει αναλυτικότερη παρουσίαση του περιβάλλοντος KODU.

Game Maker

Το **Game Maker** είναι ένα πολύ ισχυρό περιβάλλον για τη διδασκαλία του προγραμματισμού μέσω της κατασκευής ψηφιακών παιχνιδιών.

Σε επόμενη ενότητα υπάρχει αναλυτικότερη παρουσίαση του περιβάλλοντος Game Maker.

Μικρόκοσμοι προγραμματισμού

Στη wikipedia (http://en.wikipedia.org/wiki/Educational_programming_language) υπάρχει ένας πολύ πλούσιος κατάλογος γλωσσών και περιβαλλόντων προγραμματισμού που χρησιμοποιούνται ή μπορούν να χρησιμοποιηθούν για εκπαιδευτικούς σκοπούς. Επίσης, πολλές γλώσσες προγραμματισμού (μεταξύ των οποίων και εκπαιδευτικού χαρακτήρα) βρίσκονται στη διεύθυνση:

<http://progopedia.com/>

To Mortran

(http://www.wsd1.org/LTCActivities/46%20Freeware/mortran_mouse_maze.htm) ανήκει στη γενικότερη κατηγορία των Logo-like περιβαλλόντων, αλλά έχει ένα πολύ περιορισμένο πεδίο, δεδομένου ότι πρόκειται για ένα περιβάλλον στο οποίο υπάρχει μια μοναδική λειτουργία. Η αναπαριστώμενη φυσική οντότητα είναι ένα ποντίκι, το οποίο κινούμενο μέσα σε ένα λαβύρινθο, προσπαθεί να φτάσει σε ένα σημείο όπου βρίσκεται το τυρί. Η λεπτομερής περιγραφή του υπάρχει σε παράρτημα.

Άλλα εναλλακτικά περιβάλλοντα

AgentSheets

Αποτελεί ένα πολύ διαδεδομένο περιβάλλον κυρίως στις Η.Π.Α.. Μεταφράστηκε και στα ελληνικά (μια παλιότερη έκδοσή του) Σήμερα έχει αναπτυχθεί και σε 3 διαστάσεις. Είναι εμπορικό προϊόν.

Το **ToonTalk** ("Toon" πιθανότατα από το "Cartoon") αποτελεί ένα περιβάλλον για την εκμάθηση του προγραμματισμού από παιδιά βασισμένο στο animation: <http://www.toontalk.com/>. Είναι μεταφρασμένο και σε άλλες γλώσσες (δυστυχώς όχι στα ελληνικά). Διατίθεται δωρεάν (η πλήρης Αγγλική έκδοση είναι περίπου 120 MB). (Τελευταία επίσκεψη τον Αύγουστο του 2012).

Το **Phrogram** (<http://phrogram.com/>) είναι ένα εμπορικό προϊόν (διατίθεται δωρεάν για 30 μέρες δοκιμή μόνο) για μια εισαγωγή στον προγραμματισμό, με μια διαφορετική προσέγγιση. Οι δημιουργοί του αναφέρουν ότι κατά την άποψή τους, ξεκινώντας τη διδασκαλία του προγραμματισμού με Java, C++ ή Visual Basic θα έχει να αντιμετωπίσει προβλήματα με τη σύνταξη τους (που είναι δύσκολη για αρχαίους) και θα είναι υποχρεωμένος να εργάζεται σε πολύπλοκα "development environments" όπως το Visual Studio ή το Eclipse. Εξάλλου η φιλοσοφία που το διέπει είναι να δώσει στον προγραμματιστή από την αρχή μια «ιδέα» για τον προγραμματισμό κοντά στην «πραγματικότητα» και δεν επιδιώκει να «κρύψει» (όπως αναφέρεται στο site) τον πηγαίο κώδικα πίσω από τεχνικές visual ή "drag-and-drop", ούτε και να παρουσιάσει υπερ-απλουστευμένες τις

προγραμματιστικές έννοιες.
(Τελευταία επίσκεψη τον Ιούλιο του 2012).

Το **HANDS** (<http://www.cs.cmu.edu/~pane/ftp/acse-ftp/paue-ftp/ESPGradWorkshop.pdf>) είναι ένα περιβάλλον που, όπως δείχνει και το όνομά του, στηρίζεται στην ιδέα της χρηστικής διεπαφής προκειμένου να διευκολύνει τον προγραμματισμό από μικρά παιδιά. Ωστόσο δεν είναι απολύτως σαφές σε ποιο βαθμό έχει υλοποιηθεί - ωστόσο από το προαναφερόμενο κείμενο είναι δυνατόν να κατανοήσει κανείς την προβληματική του.

Το **YENKA** (http://www.yenka.com/en/Yenka_Programming/) είναι ένα περιβάλλον το οποίο εισάγει τους αρχάριους στις βασικές έννοιες του προγραμματισμού μέσα από την (προγραμματιζόμενη) animation χαρακτήρων - όπως ανθρώπων. Σε επόμενη ενότητα υπάρχει αναλυτικότερη παρουσίαση του περιβάλλοντος YENKA.

7. ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ ΚΑΙ ΕΡΓΑΣΙΕΣ

Δείτε το παρακάτω βίντεο:

Δραστηριότητα 0 http://www.youtube.com/watch?feature=player_embedded&v=nKIu9yen5nc

Δραστηριότητα 1^η

Εμβάθυνση σε έννοιες της Διδακτικής: διδακτική μετατόπιση και διδακτικό συμβόλαιο

Στο θεωρητικό τμήμα προτείνονται ορισμένες βασικές έννοιες. Μεταξύ αυτών είναι και η έννοια της διδακτικής μετατόπισης - μια έννοια που χρησιμοποιήθηκε πολύ στη Διδακτική των Μαθηματικών. Στο πλαίσιο της εξετάζονται θέματα που σχετίζονται με τη "ζωή" των διαφόρων ενοτήτων που διδάσκονται: για παράδειγμα, από τα Μαθηματικά πάντοτε, για λόγους που οφείλονται σε νέες παιδαγωγικές-διδακτικές αντιλήψεις, έννοιες, κεφάλαια και τεχνικές συρρικνώνονται (οι λεγόμενοι Γεωμετρικοί τόποι ουσιαστικά δεν διδάσκονται πλέον και η θέση της Γεωμετρίας στο Λύκειο είναι πια πολύ περιορισμένη), άλλα εξαφανίζονται τελείως (όπως η διδασκαλία της περίφημης "απλής και σύνθετης μεθόδου των τριών", τα περί μειγμάτων και κραμάτων κ.ά.), άλλων η θέση αποδυναμώνεται (κάποτε τα περί απολύτων τιμών ήταν ξεχωριστό κεφάλαιο για τις Πανελλαδικές εξετάσεις), άλλα μετατοπίζονται προς "τα κάτω" (στο Δημοτικό, εδώ και πολλά

χρόνια διδάσκονται εξισώσεις, πράξεις με παρενθέσεις κλπ - θέματα που διδασκόταν αποκλειστικά στο Γυμνάσιο).

Είναι σαφές πως και η ύλη των Πανελλαδικών εξετάσεων επηρεάζεται από τα φαινόμενα αυτά. Υπάρχουν επίσης παραδείγματα αντικειμένων που διδάσκονται με τρόπους οι οποίοι δεν έχουν αναφορά στο αντίστοιχο αντικείμενο των Μαθηματικών: για παράδειγμα, οι δεκαδικοί, μερικές φορές παρουσιάζονται ως "φυσικοί με υποδιαστολή". Για όσες και όσους ενδιαφέρονται για περαιτέρω μελέτη, υπάρχει μια ελληνική Διδακτορική διατριβή με θέμα την διδακτική μετατόπιση στη διδασκαλία της απόλυτης τιμής. (<http://thesis.ekt.gr/thesisBookReader/id/3660#page/8/mode/2up>)

Η πρόοδος της Επιστήμης στα Μαθηματικά δεν επηρεάζει σημαντικά τη διδασκαλία τους, αφού τα διδασκόμενα αντικείμενα στα Μαθηματικά είναι "ηλικίας" πολλών δεκαετιών - αν όχι χιλιετιών. Αντίθετα, σε μαθήματα όπως η Βιολογία, η ταχύτατη πρόοδος των τελευταίων ετών υποχρεώνει το εκπαιδευτικό σύστημα σε αναμόρφωση των διδακτέας ύλης, ενώ η μεταβολή των κοινωνικών πρακτικών και απόψεων, ακόμη και σε θέματα ηθικής (όπως η Βιοηθική) επηρεάζει το σχολικό πρόγραμμα. Έχουμε άραγε ανάλογα φαινόμενα στην Πληροφορική; Δηλαδή υπάρχουν κεφάλαια, έννοιες, τεχνικές, θέματα γενικά που έχουν υποστεί κάποια μεταβολή στη σχολική "ζωή" τους στο χρόνο; Αν κρίνετε πως "ΝΑΙ" ποια μπορεί να είναι τα ενδεχόμενα αίτια; Η πρόοδος της Επιστήμης και του επαγγέλματος, η αλλαγή στις κοινωνικές πρακτικές, η μεταβολή των διδακτικών/παιδαγωγικών απόψεων της "νοόσφαιρας" ή γενικότερα η μεταβολή στα διδακτικά μοντέλα γενικώς; Η ύλη και το "στυλ" των θεμάτων στις Πανελλαδικές εξετάσεις επηρεάζονται από αυτό; Συζητήστε το σχετικό θέμα με τους συναδέλφους σας.

Δραστηριότητα 2^η

**Μελέτη του
θεωρητικού
μέρους**

Μελετήστε το θεωρητικό μέρος και επισημάνετε τους ορισμούς βασικών εννοιών: διδακτικό σύμβολο, διδακτική κατάσταση, διδακτική μετατόπιση. Θεωρείτε ότι οι έννοιες αυτές έχουν κάποια χρησιμότητα για τον εκπαιδευτικό (οιουδήποτε αντικειμένου) ή είναι άχρηστες στη διδακτική πράξη; Αν η απάντησή σας είναι "ΝΑΙ",

τότε ποια νομίζετε ότι είναι η χρησιμότητά τους; Ειδικότερα, θεωρείτε ότι έχουν κάποια χρησιμότητα στη διδασκαλία της Πληροφορικής; Συζητήστε το θέμα αυτό στο forum (σύγχρονα ή ασύγχρονα). Συντάξτε μια απάντηση μετά τη συζήτηση και υποβάλετε την (η εργασία-απάντηση περίπου 500 λέξεις με τη συνήθη τυπική μορφοποίηση).

Επισκεφθείτε τη σχετική σελίδα της Wikipedia:

http://en.wikipedia.org/wiki/Educational_programming_language

Δραστηριότητα 3^η
**Εκπαιδευτικές
γλώσσες
προγραμματισμού**

και δείτε τις κατηγορίες "εκπαιδευτικών" γλωσσών προγραμματισμού που αναφέρονται (ο κατάλογος δεν είναι βεβαίως πλήρης). Θεωρείτε ότι ο αριθμός των γλωσσών αυτών που έχουν εκπαιδευτικό χαρακτήρα είναι υπερβολικά μεγάλος; Εκτιμάτε ότι κάποιες από τις γλώσσες αυτές υπερτερούν έναντι των υπολοίπων; Ποια είναι η γνώμη σας γενικότερα για τις κατηγορίες αυτές των γλωσσών και περιβαλλόντων;

Στη σχετική σελίδα της Wikipedia:

http://el.wikipedia.org/wiki/Γλώσσα_προγραμματισμού

υπάρχει μια σχετικά σύντομη αναφορά στις κυριότερες γλώσσες προγραμματισμού και στη διεύθυνση:
<http://el.wikipedia.org/wiki/Αλγόριθμος> μια παρουσίαση της έννοιας του αλγορίθμου (με αρκετές παραπομπές).

Δραστηριότητα 4^η
**Προγραμματισμός
και γλώσσες
προγραμματισμού**

Στο ψηφιακό σχολείο (<http://digitalschool.minedu.gov.gr/>) μπορείτε να δείτε το σχετικό κεφάλαιο για τον προγραμματισμό της Γ' Γυμνασίου:
(http://digitalschool.minedu.gov.gr/modules/ebook/show.php/DSB102/365/2448,9367/index1_1.html).

Οδηγίες για τον καθηγητή υπάρχουν στο:

http://digitalschool.minedu.gov.gr/modules/document/file.php/DSB102/Διδακτικό_Πακέτο/Βιβλίο_Εκπαιδευτικού/Γ_Γυμνασίου/1.pdf

Δείτε επίσης τις προτάσεις του σχολικού βιβλίου για την έννοια του αλγορίθμου και του προγράμματος. Κατά τη γνώμη σας τα παραδείγματα που δίνονται είναι κατάλληλα για να σχηματίσει ο μαθητής την ιδέα ενός αλγορίθμου και ενός προγράμματος;

Μια πολύ ιδιαίτερη «γλώσσα προγραμματισμού» προσφέρεται στο ψηφιακό σχολείο:

http://digitalschool.minedu.gov.gr/modules/ebook/show.php/DSB102/365/2448,9367/extras/Tools-Applications/Kef1_1_MinilDE_simple/Kef1_1_MinilDE_simple.html

(προαπαιτεί εγκατάσταση java).

Το σχετικό ΔΕΠΠΣ-ΑΠΣ υπάρχει στη διεύθυνση:

http://digitalschool.minedu.gov.gr/modules/document/file.php/DSB101/ΔΕΠΠΣ-ΑΠΣ_Πληροφορικής/18deppsaps_Pliroforikis.pdf

Τέλος, τα σχετικά κείμενα για τα νέα πιλοτικά προγράμματα δημοτικού και Γυμνασίου βρίσκονται στο:

<http://digitalschool.minedu.gov.gr/info/newps.php>

Επισκεφθείτε τις ιστοσελίδες της wikipedia που αναφέρονται στον CARDIAC και δείτε το σχετικό φιλμάκι στο YouTube. Ένας «H.Y.» που λειτουργεί με το χέρι φαίνεται να ανήκει στο μουσείο των αξιοπερίεργων αντικειμένων και επιπλέον η διδακτική του χρησιμότητα δεν είναι προφανής.. Αυτοί που επινόησαν τον CARDIAC ωστόσο και τον διέδωσαν (δείτε και το αντίστοιχο Γαλλικό κατασκεύασμα με τον τίτλο Ordinaroche), θεώρησαν ότι αυτή η προσομοίωση, από την ίδια της τη φύση καθιστά πολύ εύκολη την κατανόηση του τρόπου με τον οποίο λειτουργεί ένας πραγματικός H.Y. (τουλάχιστον μερικές από τις λειτουργίες του). Η «ιδέα» δηλαδή πίσω από την επινόηση του συστήματος αυτού είναι η διδασκαλία. Ποια θα μπορούσε να είναι η διδακτική χρησιμότητα ενός τέτοιου συστήματος; Σας φαίνεται ότι το σύστημα αυτό θα μπορούσε να έχει ακόμη και σήμερα κάποια διδακτική χρησιμότητα; Σε ποιες περιπτώσεις θα μπορούσε να είναι χρήσιμο; Σε ορισμένα άλλα σημεία των σημειώσεων, προτείνονται μερικές εφαρμογές στην τάξη στις οποίες οι μαθητές προσομοιώνουν τη λειτουργία ενός τμήματος ενός H.Y. παίζοντας οι ίδιοι το ρόλο των διαφόρων μερών του H.Y. Υπάρχει κάποια ομοιότητα ανάμεσα στα δυο «συστήματα»: τον CARDIAC και την προσομοίωση αυτή;

Δραστηριότητα 5^η

CARDIAC

Δραστηριότητα 6^η

Επισκεφθείτε τη σχετική σελίδα της γλώσσας ή περιβάλλοντος

- BALTIE** BALTIE-3 <http://progopedia.com/language/baltie/>
Ποια είναι η γνώμη σας για ένα περιβάλλον αυτού του είδους;
- Κατεβάστε το λογισμικό Mortran (δεν χρειάζεται εγκατάσταση). Ανοίξτε το και προσπαθήστε να δημιουργήσετε και να εκτελέσετε 2-3 απλά προγράμματα. Τα χαρακτηριστικά του Mortran (απλό περιβάλλον, απουσία ιδιαίτερης σύνταξης) σας φαίνονται ικανοποιητικά για μια εισαγωγή στον προγραμματισμό ή θεωρείτε ότι το περιβάλλον αυτό είναι μάλλον "φτωχό" σε διδακτικές δυνατότητες και δεν είναι χρήσιμο; Σας φαίνεται περισσότερο παιχνίδι παρά πραγματικά κατάλληλο για μια στοιχειώδη εισαγωγή τον προγραμματισμό; Στο παράρτημα 3 των σημειώσεων υπάρχει μια λεπτομερέστερη περιγραφή του περιβάλλοντος, καθώς και μερικές οδηγίες για τον τρόπο με τον οποίο θα μπορούσε να χρησιμοποιηθεί στην τάξη.
- Δραστηριότητα 7^η
Το περιβάλλον MORTRAN
- Μελετήστε την παράγραφο 5. Κατά πάσα πιθανότητα από προσωπική εμπειρία θα έχετε χρησιμοποιήσει τουλάχιστον μια γλώσσα "επαγγελματική" (όπως Java, C, Visual BASIC παλιότερα Pascal ή FORTRAN) και ίσως έχετε εμπειρία και από συστήματα (όπως IDE) επαγγελματικού χαρακτήρα. Στην παράγραφο 5 περιγράφονται ορισμένες διαφορές ανάμεσα στα δυο είδη περιβαλλόντων (αν και η διάκριση δεν είναι πάντοτε εύκολη ή εφικτή). Συμφωνείτε με τις απόψεις που εκφράζονται στην παράγραφο 5;
- Δραστηριότητα 8^η
Διαφορές ανάμεσα στα επαγγελματικά και τα εκπαιδευτικά περιβάλλοντα
- Επισκεφθείτε τον ιστοχώρο του AgentSheet. Περιηγηθείτε στον ιστοχώρο του. Αν και το AgentSheet είναι ένα εμπορικό προϊόν, το ελληνικό δημόσιο χρηματοδότησε ορισμένες εφαρμογές στο πλαίσιο του έργου "Πλειάδες". Στο <http://www.agentsheets.gr/> μπορείτε να βρείτε μερικά applets που έχουν δημιουργηθεί με το περιβάλλον αυτό. Δείτε μερικές εφαρμογές. Θα ήταν εφικτό να δημιουργηθούν εφαρμογές προσομοίωσης λειτουργίας συστημάτων (δικτύων, λειτουργία του επεξεργαστή κ.ά.); Συζητείστε αυτή τη δυνατότητα.
- Δραστηριότητα 9^η
AgentSheet

Δραστηριότητα 10 ^η Game Maker	Επισκεφθείτε τον ιστοχώρο του Game Maker. Περιηγηθείτε στον ιστοχώρο του. Επισημάνετε τα χαρακτηριστικά του περιβάλλοντος και της γλώσσας προγραμματισμού: σας φαίνεται ότι είναι κατάλληλη για μαθητές, αρχάριους προγραμματιστές.; Διακρίνετε κάποια πλεονεκτήματα-μειονεκτήματα σε σχέση με τα άλλα περιβάλλοντα;
Δραστηριότητα 11 ^η Kodu	Επισκεφθείτε τον ιστοχώρο του Kodu Περιηγηθείτε στον ιστοχώρο του. http://research.microsoft.com/en-us/projects/kodu/ Επισημάνετε τα χαρακτηριστικά του περιβάλλοντος και της γλώσσας προγραμματισμού: σας φαίνεται ότι είναι κατάλληλη για μαθητές, αρχάριους προγραμματιστές; Διακρίνετε κάποια πλεονεκτήματα-μειονεκτήματα σε σχέση με τα άλλα περιβάλλοντα;
Δραστηριότητα 12 ^η Ο «χάρτινος» υπολογιστής	Μελετήστε το "χάρτινο" υπολογιστή του παραρτήματος 1. Αντιστοιχεί σε διδακτικές πρακτικές που χρησιμοποιείτε και εσείς; Ποια είναι η γνώμη σας για τον "Η.Υ." αυτό (δηλαδή ποια είναι κατά τη γνώμη σας, τα ενδεχόμενα προτερήματα και μειονεκτήματά του);
Δραστηριότητα 13 ^η YENKA	Επισκεφθείτε τον ιστοχώρο του YENKA (http://www.yenka.com/). Το περιβάλλον YENKA προτείνει μια εισαγωγή στον προγραμματισμό βασισμένη στη διαχείριση (από το χρήστη) μια σειράς animated οντοτήτων (χαρακτήρων). Περιηγηθείτε στον ιστοχώρο του και εξετάστε μερικές από τις κατασκευές που προτείνονται για επίδειξη. Επισημάνετε τα χαρακτηριστικά του περιβάλλοντος και της γλώσσας προγραμματισμού: σας φαίνεται ότι είναι κατάλληλη για μαθητές, αρχάριους προγραμματιστές; Διακρίνετε κάποια πλεονεκτήματα-μειονεκτήματα σε σχέση με τα άλλα περιβάλλοντα;
Δραστηριότητα 14 ^η Προσομοίωση εκτέλεσης ενός προγράμματος	Πολύ πιο «κοντά» στη λειτουργία ενός Η.Υ. είναι η δραστηριότητα – ή ακριβέστερα το σύνολο των δραστηριοτήτων – στις οποίες η τάξη προσομοιώνει τη λειτουργία ενός Η.Υ. κατά την εκτέλεση ενός προγράμματος. Η τάξη πρέπει να συμμετέχει στο σύνολο της ή, εν πάση περιπτώσει, κατά το μεγαλύτερο δυνατόν τμήμα της. Η

κεντρική ιδέα είναι ότι η εκτέλεση ενός προγράμματος θα είναι κατακερματισμένη – αφού κάθε μαθητής θα εκτελεί ένα μόνο τμήμα της όλης διαδικασίας. Κατά πάσα πιθανότητα εξ άλλου, οι μαθητές δεν θα έχουν μια συνολική εικόνα του προγράμματος. Παρόλα αυτά, το πρόγραμμα θα εκτελεστεί και θα παραχθεί κάποιο αποτέλεσμα. Ή δραστηριότητα αυτή θέλει προσεκτική προετοιμασία: πρέπει να επιλεγούν προσεκτικά οι «εντολές» της «γλώσσας προγραμματισμού» που θα χρησιμοποιηθεί. Στην προκειμένη περίπτωση εξ άλλου, θα μπορούσε να δημιουργηθεί μια παραλλαγή μιας απλής γλώσσας προγραμματισμού ή ένα ελληνικό ισοδύναμο της.

Βασικά στοιχεία του μηχανισμού του Η.Υ. είναι η «μονάδα εισόδου» η «μονάδα εξόδου», ο «επεξεργαστής», η «μνήμη» και το «λεωφορείο». Η «μονάδα εισόδου» μπορεί να είναι ένας μαθητής στον μαυροπίνακα, ο οποίος να σημειώνει τους αριθμούς που του δίνονται από το «λεωφορείο» γραμμένοι σε ένα χαρτί. Παρόμοια, η «μονάδα εξόδου» αναγράφει στο μαυροπίνακα αριθμούς που τους παραλαμβάνει γραμμένους σε ένα χαρτί. Η μνήμη – ανάλογα με τον αριθμό των μαθητών που συμμετέχουν - μπορεί να αποτελείται από πολλούς μαθητές. Το ρόλο του «λεωφορείου» τον παίζει ένας μαθητής που μεταφέρει γραπτή μηνύματα και ο «επεξεργαστής» τέλος εκτελεί τις αναγκαίες πράξεις και δίνει εντολές στο «λεωφορείο» για τη διακίνηση των δεδομένων, αποτελεσμάτων κλπ. Βασικό κέρδος από τη δραστηριότητα αυτή είναι η κατανόηση μερικών βασικών αρχών λειτουργίας ενός Η.Υ. σε σχέση με την εκτέλεση ενός προγράμματος. Τα τμήματα του «υπολογιστή» συνεργάζονται, εκτελούν το καθένα ένα μικρό τμήμα της δουλειάς και παράγουν ένα αποτέλεσμα – χωρίς καν να γνωρίζουν ποιο είναι το νόημα του αποτελέσματος.

Δραστηριότητα
15^η

**Online
περιβάλλοντα για
την εκμάθηση του
προγραμματισμού**

Υπάρχουν στο Διαδίκτυο πολλά περιβάλλοντα για την εκμάθηση του προγραμματισμού.

Δύο τέτοια αξιοσημείωτα περιβάλλοντα (στην Αγγλική γλώσσα και τα δύο) είναι το Codecademy (<http://www.code.org/learn/codecademy>) και το w3schools:

<http://www.w3schools.com>

Και τα δύο περιβάλλοντα προσφέρουν δωρεάν μαθήματα είτε σε markup γλώσσες (όπως η html) , είτε σε τυπικές γλώσσες προγραμματισμού, όπως η JavaScript.

Παρόμοιο είναι και το περιβάλλον Code Avengers:

<http://www.codeavengers.com/>

Ακόμη, στα περιβάλλοντα που παρέχουν πληροφορίες αυτού του είδους (Wikipedia

http://en.wikipedia.org/wiki/List_of_educational_programming_languages, Youtube,...) μπορεί κανείς να βρει πολλές πληροφορίες ή

σχετικά μαθήματα – π.χ. στο YouTube οι λέξεις-κλειδιά learning programming ή learning programming languages παραπέμπουν σε πολλά βίντεο όπου κανείς μπορεί να παρακολουθήσει μαθήματα για Visual Basic, C κλπ.

Μαθήματα προγραμματισμού που απευθύνονται σε άτομα που θέλουν να μάθουν προγραμματισμό σε πιο προηγμένο επίπεδο προσφέρουν και τα περιβάλλοντα μαθημάτων online Coursera (<https://www.coursera.org/>), Udacity

(<https://www.udacity.com/courses>) και γενικότερα τα ανοιχτά μαθήματα που ανήκουν στην κατηγορία MOOC (Massively Open Online Courses) κ.ά. Μια ικανοποιητική λίστα με MOOC, μαζί με ένα επεξηγηματικό βίντεο είναι στη διεύθυνση: http://www.bdpa-detroit.org/portal/index.php?option=com_content&view=article&id=57:moocs-top-10-sites-for-free-education-with-elite-universities&catid=29:education&Itemid=20

Δραστηριότητα
16^η

**Η μηχανή του
Post**

Πρόκειται για μια «θεωρητική μηχανή» (που «λειτουργεί» στο χαρτί αν και υπάρχουν προσομοιώσεις της) με τους απλούστερους δυνατούς κανόνες.

Περιγράφεται στο παράρτημα 2.

Ερωτήσεις "κλειστού τύπου":

Σε ένα προγραμματιστικό μικρόκοσμο η γλώσσα προγραμματισμού αποτελείται συνήθως από ένα περιορισμένο ρεπερτόριο εντολών με απλή σύνταξη και σημασιολογία.

ΣΩΣΤΟ ΛΑΘΟΣ

Η πλειοψηφία των μικρόκοσμων προγραμματισμού ενσωματώνει ένα ανοιχτό περιβάλλον που βασίζεται σε κάποιο φυσικό μοντέλο, ενώ ο χρήστης ελέγχει ένα πρωταγωνιστή που «ζει» στο περιβάλλον αυτό.

ΣΩΣΤΟ ΛΑΘΟΣ

Τα προβλήματα που καλούνται να λύσουν οι σπουδαστές σε ένα προγραμματιστικό μικρόκοσμο είναι κυρίως προβλήματα επεξεργασίας αριθμών και συμβόλων.

ΣΩΣΤΟ. Οι προγραμματιστικοί μικρόκοσμοι έχουν ως βασικό στόχο τους την εξοικείωση των αρχαρίων προγραμματιστών με τις βασικές έννοιες του προγραμματισμού. Έτσι, τα προβλήματα επεξεργασίας αριθμών και συμβόλων είναι τα πλέον κατάλληλα στα εισαγωγικά μαθήματα – λόγω της απλότητας στο «χειρισμό» τους.

ΛΑΘΟΣ. Οι προγραμματιστικοί μικρόκοσμοι έχουν ως βασικό στόχο τους την εξοικείωση των αρχαρίων προγραμματιστών με τις βασικές έννοιες του προγραμματισμού. Για το λόγο αυτό, κατά κανόνα, οι προγραμματιστικοί μικρόκοσμοι στηρίζονται σε περιβάλλοντα «διαχείρισης» διαφόρων οντοτήτων, έτσι ώστε οι υπό διδακτική διαπραγμάτευση έννοιες να είναι πιο απλές, πιο οικείες και πιο ενδιαφέρουσες για τους αρχάριους μαθητές.

Σε ένα προγραμματιστικό μικρόκοσμο παρουσιάζεται, συνήθως, το τελικό αποτέλεσμα της εκτέλεσης ενός προγράμματος και δεν υποστηρίζεται η βήμα προς βήμα εκτέλεση.

ΣΩΣΤΟ ΛΑΘΟΣ

8. ΧΡΗΣΗ ΕΞΩΤΕΡΙΚΩΝ ΠΗΓΩΝ

ΒΙΒΛΙΟΓΡΑΦΙΑ (ενδεικτική)

1. Anderson P., Seaquist C.R. (1998) *Using StarLogo to Introduce Differential Equations*

- 12th-Annual International Conference on Technology in Collegiate Mathematics 1998.
2. Arsac J. (1983) *Les bases de la programmation*, Dunod.
 3. Baecker, R. (1997) *Sorting Out Sorting: A Case Study of Software Visualization for Teaching Computer Science*, In *Software Visualization: Programming as a multimedia Experience*, Stasko, J., Domingue, J., Brown, M. and Price, B. (Eds.), MIT Press, Cambridge, pp. 369-381.
 4. Barnes J. D. (2002) *Teaching Introductory Java through LEGO Mindstorms Models*, *Proceedings of the 33rd SIGCSE Technical Symposium on Computer Science Education*.
 5. Bayman P., Mayer R. E.,(1983) *A Diagnosis of Beginning Programmers' Misconceptions of BASIC Programming Statements*, *Communications of the ACM*, vol. 26, n° 9, pp. 677-679.
 6. Beaudoin C., Meyer G. (1981) *Méthodes de programmation*, Eyrolles
 7. Whitley K. N. (1997) *Visual programming languages and the empirical evidence for and against*. *Journal of Visual Languages and Computing*, 5(1), pp. 109–142.
 8. Wick M. (1995) *On Using C++ and Object-Orientation in CS1: the Message is still more important than the Medium*, *ACM SIGCSE Bulletin*, Vol. 27, Issue 1, 322-326.
 9. Xinogalos S. & Satratzemi M. (2004), *Studying Novice Programmers' Attitudes in Developing and Implementing Algorithms Using an Educational Programming Environment*, *Proceedings of the International Conference on Cybernetics and Information Technologies, Systems and Applications (CITSA 2004)*, Orlando, Florida, USA, July 21-25, Vol. 1, 198-203.
 10. Xinogalos S., Sartatzemi M., Dagdilelis V., (2006a) «*Studying Students' Difficulties in an OOP Course based on BlueJ*», *IATED International Conference on Computers and Advanced technology in Education (CATE2006)*, October 4-6, 2006, Lima, Peru, 82-87.
 11. Xinogalos S., Sartatzemi M., Dagdilelis V., (2006b), «*Evaluating objectKarel - an educational programming environment for object oriented programming*», *IV International Conference on Multimedia and ICTs in Education (m-ICTE2006)*, Seville, Spain, 22-25 November 2006, 821-825.
 12. Xinogalos S., Satratzemi M., (2003), «*Students' Practices in Developing and Implementing Algorithms: An Empirical Study*», *proceedings of the 6th Hellenic European Conference on Computer Mathematics and its Applications (HERCMA 2003)*, 754-763.
 13. Xinogalos S., Satratzemi M., (2004a), "*Introducing Novices to Programming: a review of Teaching Approaches and Educational Tools*", *International Conference on Education and Information Systems, Technologies and Applications (EISTA 2004)*, Orlando, USA, on July 21-25, 2004, 60-65.
 14. Xinogalos S., Satratzemi M., (2004b), "*Studying Novice Programmers' Attitudes in Developing and Implementing Algorithms Using an Educational Programming Environment*", *proceedings of the International Conference on Cybernetics and Information Technologies, Systems and Applications (CITSA 2004)*, Orlando, USA, on July 21-25, 2004, 198-203.
 15. Xinogalos S., Satratzemi M., Dagdilelis V., (2003), «*An environment for Teaching Object-Oriented Programming: ObjectKarel*», *3rd IEEE International Conference on Advanced*

- Learning Technologies (EEEE ICALT 2003), July 9-11, 2003, Athens, Greece, 342-343.*
16. Xinogalos S., Satratzemi M., Dagdilelis V., (2006a), «An Introduction to Object-oriented programming with a didactic Microworld: objectKarel», *Computers & Education*, 47, 2006, 148-171, Elsevier Publishers.
 17. Xinogalos S., Satratzemi M., Dagdilelis V., (2006b), «An Objects-First Approach to Teaching Object Orientation based on objectKarel», *5th WSEAS International Conference on Education and educational Technology (EDU'06), Tenerife, Spain, 16-18 December 2006, 93-98.*
 18. Xinogalos S., Satratzemi M., Dagdilelis V., (2006c), «Teaching Fundamental Notions of Object Oriented Programming with objectKarel», *WSEAS TRANSACTIONS on ADVANCES in ENGINEERING EDUCATION*, 11, 3, 1022-1029.
 19. Xinogalos S., Satratzemi M., Dagdilelis V., (2007a), "Teaching Java with BlueJ: a Two-Year Experience", poster presentation in *ITICSE 2007, Dundee, Scotland, 25-27 June 2007.*
 20. Xinogalos S., Satratzemi M., Dagdilelis V., (2007b), "Re-designing an OOP course based on BlueJ", *The 7th IEEE International Conference on Advanced Learning Technologies (EEEE ICALT 2007), July 18-20, 2007, Niigata, Japan.*
 21. Xinogalos S., Satratzemi M., Dagdilelis V., Evangelidis G., (2006d), "Teaching OOP with BlueJ: A Case Study", *6th IEEE International Conference on Advanced Learning Technologies (EEEE ICALT 2006), July 5-7, 2006, Kerkrade, The Netherlands, 944-946.*
 22. Xinogalos Stelios, Satratzemi Maya and Dagdilelis Vassilios, *An introduction to objectoriented programming with a didactic microworld: objectKarel, Computers & Education, Elsevier, In Press (Available online 8 December 2004)*
 23. Xinogalos, S. & Satratzemi, M. (2002) *An Integrated Programming Environment for Teaching the Object-Oriented Programming Paradigm. Accepted for presentation in the 1st EurAsian Conference on Advances in Information and Communication Technology and will be published in "Lecture Notes in Computer Science" by Springer Verlag, Shiraz, Iran, October 2002.*
 24. Xinogalos, S. & Satratzemi, M. (2004), *Studying Novice Programmers' Attitudes in Developing and Implementing Algorithms Using an Educational Programming Environment, Proceedings of the 10th International Conference on Information Systems Analysis and Synthesis (ISAS 2004) jointly with the International Conference on Cybernetics and Information Technologies, Systems and Applications (CITSA 2004), Orlando, Florida, USA, July 21-25, Vol. 1, 198-203.*
 25. Xinogalos, S. & Satratzemi, M. (2005a), *Using Hands-on Activities for Motivating Students with OOP Concepts Before They Are Asked to Implement Them, Proceedings of the 10th annual Conference on Innovation and Technology in Computer Science Education.*
 26. Xinogalos, S. & Satratzemi, M. (2005b), *The Hands-on Activities of the Programming Microworld objectKarel, Proceedings of the 10th annual Conference on Innovation and Technology in Computer Science Education.*
 27. Xinogalos, S., and Satratzemi, M. (2005a), *The Hands-on Activities of the Programming Microworld objectKarel, In Proceedings of the 10th annual ACM Conference on*

- Innovation and Technology in Computer Science Education (ITiCSE 2005)*, 384
28. Xinogalos, S., and Satratzemi, M. (2005b), *Using Hands-on Activities for Motivating Students with OOP Concepts Before They Are Asked to Implement Them*, In *Proceedings of the 10th annual Conference on Innovation and Technology in Computer Science Education (ITiCSE 2005)*, 380
 29. Xinogalos, S., Satratzemi, M. & Dagdilelis, V. (2005c), *An Introduction to object-oriented programming with a didactic microworld: objectKarel*, Accepted for publication in *Computers & Education*, Elsevier Publishers.
 30. Xinogalos, S., Satratzemi, M. (2002), *An Integrated Programming Environment for Teaching the Object-Oriented Programming Paradigm*, *Lecture Notes in Computer Science (LNCS)*, 2510, EURASIA-ICT 2002, Shafazand H A Min Tjoa (Eds.), Springer Verlag, 544-551.
 31. Xinogalos, S., Satratzemi, M. (2002). "An Integrated Programming Environment for Teaching the Object-Oriented Programming Paradigm". *EurAsia-ICT 2002*: 544-551
 32. Xinogalos, S., Satratzemi, M. and Dagdilelis, V. *An introduction to object-oriented programming with a didactic microworld: objectKarel*, *Computers & Education*, Volume 47, Issue 2, September 2006, 148-171
 33. Ziegler U. & Crews T. (1999) *An Integrated Program Development Tool for Teaching and Learning How to Program*, *ACM SIGSCE Bulletin*, Vol. 31, No. 1, pp. 276-280.
 34. Βίγκλας Λ & Σπυρόπουλος Χ. (2003) *Δημιουργία Προσομοιώσεων με Εκπαιδευόμενους Πράκτορες: "Η Εκπαιδευτική αξία της κατασκευής Προσομοιώσεων από τους μαθητές στο περιβάλλον εκπαιδευόμενων πρακτόρων AgentSheets"*
 35. Δαγδιλέλης Β., (1996) *Διδακτική της πληροφορικής. Η διδασκαλία του προγραμματισμού: αντιλήψεις των σπουδαστών για την κατασκευή και επικύρωση προγραμμάτων και διδακτικές καταστάσεις για τη διαμόρφωσή τους*, *Διδακτορική διατριβή, Τμήμα Εφ. Πληροφορικής Πανεπιστήμιο Μακεδονίας*.
 36. Δαγδιλέλης, Β. & Σατρατζέμη, Μ. (1997) *Η μηχανή του Post: ένας διδακτικός μικρόκοσμος για την εισαγωγή στον τυπικό προγραμματισμό. Πρακτικά 6ου Πανελληνίου Συνεδρίου Πληροφορικής, Αθήνα, Δεκέμβριος 1997, σελ. 193-203*.
 37. Δαγδιλέλης, Β. & Σατρατζέμη, Μ. (1998) *Λογισμικό για την διδασκαλία και την έρευνα στη θεωρία γραφημάτων: αλγοριθμικά και διδακτικά προβλήματα.*, 12ο Εθνικό Συνέδριο Επιχειρησιακής Έρευνας (Ε.Ε.Ε.), Σάμος, Σεπτέμβριος 1998 1231-1244.
 38. Δαγδιλέλης, Β. (1996), *Διδακτική της πληροφορικής. Η διδασκαλία του προγραμματισμού: αντιλήψεις των σπουδαστών για την κατασκευή και επικύρωση προγραμμάτων και διδακτικές καταστάσεις για τη διαμόρφωσή τους*, *Διδακτορική διατριβή, Τμήμα Εφ. Πληροφορικής Πανεπιστήμιο Μακεδονίας*.
 39. Δαγδιλέλης, Β., Ευαγγελίδης, Γ., Σατρατζέμη, Μ., Εφόπουλος, Β., Κολοβός, Φ., (2000). "Το Περιβάλλον Χ: Ένας Μικρόκοσμος για μια Εισαγωγή στον Προγραμματισμό", *Πρακτικά σε ηλεκτρονική μορφή (CD-ROM) του Συνεδρίου Πληροφορική και Εκπαίδευση (Σ.Ε.Π.Δ.Ε.Θ.)*, Θεσσαλονίκη 11-12 Νοεμβρίου.
 40. Εφόπουλος Β. (2005) *Διαδικτυακό Εκπαιδευτικό Λογισμικό με υποστήριξη Βάσεων*

Δεδομένων για τη διδασκαλία της πληροφορικής σε αρχαρίους, Διδακτορική Διατριβή στο Πανεπιστήμιο Μακεδονίας, Τμήμα Εφαρμοσμένης Πληροφορικής. Θεσσαλονίκη.

41. Εφόπουλος Β., Ευαγγελίδης Γ., Δαγδιλέλης Β., Κλεφτοδήμος Αλ. (2005) *Οι Δυσκολίες των Αρχάριων Προγραμματιστών, 3ο Συνέδριο Διδακτικής της Πληροφορικής, Κόρινθος.*
42. Εφόπουλος, Β., Δαγδιλέλης, Β., Σατρατζέμη, Μ., Ευαγγελίδης, Γ. (2001b). "XWeb-Compiler - Ένα Διαδικτυακό Προγραμματιστικό Περιβάλλον με υποστήριξη Βάσεων Δεδομένων για τη διδασκαλία των αρχών του προγραμματισμού", Πρακτικά του συνεδρίου «Η Πληροφορική στην Εκπαίδευση: Τεχνολογίες, Εφαρμογές, Κατάρτιση Εκπαιδευτικών», σελ. 307-314, Ρόδος 14-15 Δεκεμβρίου.
43. Εφόπουλος, Β., Δαγδιλέλης, Β., Σατρατζέμη, Μ., Κασκάλης, Θ., Ευαγγελίδης, Γ., (2001a). "DBSchoolLab: Ένα σύστημα για τη διαχείριση ασκήσεων και διαγωνισμάτων στο σχολικό περιβάλλον", Πρακτικά του Πανελληνίου Συνεδρίου με Διεθνή Συμμετοχή «Νέες Τεχνολογίες στην εκπαίδευση και στην εκπαίδευση από απόσταση», σελ. 496-508, Ρέθυμνο, 8-10 Ιουνίου.
44. Εφόπουλος, Β., Ευαγγελίδης, Γ., Δαγδιλέλης, (2004). "Ένα περιβάλλον για τη συστηματική διδασκαλία του προγραμματισμού σε αρχαρίους", 4ο Πανελλήνιο Συνέδριο με Διεθνή Συμμετοχή "Οι τεχνολογίες της πληροφορίας και της επικοινωνίας στην εκπαίδευση" Αθήνα 29 Σεπτεμβρίου - 3 Οκτωβρίου.
45. Εφόπουλος, Β., Ευαγγελίδης, Γ., Δαγδιλέλης, Β., Κασκάλης, Θ., (2003). "Ένα μοντέλο διαδικτυακού περιβάλλοντος με υποστήριξη Βάσεων Δεδομένων για τη διδασκαλία των Αρχών του Προγραμματισμού", Πρακτικά 2ου Πανελληνίου Συνεδρίου των Εκπαιδευτικών για τις ΤΠΕ "Αξιοποίηση των Τεχνολογιών της Πληροφορίας και της Επικοινωνίας στη Διδακτική Πράξη", σελ. 733-741, Σύρος 9-11 Μαΐου.

ΔΙΚΤΥΟΓΡΑΦΙΑ

Βασική πηγή αποτελεί η Wikipedia

ΙΣΤΟΧΩΡΟΙ

Ο ιστοχώρος

<http://regulargeek.com/2011/07/20/36-resources-to-help-you-teach-kids-programming/>

περιέχει πληροφορίες κυρίως για εναλλακτικές γλώσσες προγραμματισμού για παιδιά – αν και ορισμένες από τις προτεινόμενες επιλογές δημιουργούν κάποια ερωτηματικά, όπως η πρόταση για τη PHP ή JavaScript... (τελευταία επίσκεψη τον Αύγουστο του 2012).

9. ΠΑΡΑΡΤΗΜΑ 1: Ο ΧΑΡΤΙΝΟΣ ΥΠΟΛΟΓΙΣΤΗΣ

Ο υπολογιστής παρουσιάζεται ως μια πολύ απλή μηχανή που εκτελεί υπολογισμούς, η λειτουργία της οποίας μπορεί να προσομοιωθεί στο χαρτί (από εκεί κι η ονομασία). Ο χάρτινος υπολογιστής είναι λοιπόν φορητός, οικολογικός (δεν καταναλίσκει καμιά ενέργεια)

και δεν κινδυνεύει από ιούς.

Σχετικές έρευνες (δες προηγούμενες παραγράφους) δείχνουν ότι μια εισαγωγή στον προγραμματισμό με τη βοήθεια τέτοιων νοητικών εργαλείων μπορεί πολύ εύκολα να αποτελέσει πηγή διδακτικών εμποδίων και να δυσχεράνει την εξέλιξη του σπουδαστή. Ωστόσο η εισαγωγή στην αλγοριθμική με τη βοήθεια του χάρτινου υπολογιστή είναι ιδιαίτερα εύκολη και για το λόγο αυτό πολλές φορές επιλέγεται, παρά τον κίνδυνο να δημιουργήσει “κακές προγραμματιστικές συνήθειες”. Στο συνηθισμένο σχολικό περιβάλλον της δευτεροβάθμιας εκπαίδευσης, στο πλαίσιο μιας γενικής εισαγωγής στον προγραμματισμό, ο σκοπός της διδασκαλίας του αντίστοιχου μαθήματος δεν είναι βέβαια η κατάρτιση επαγγελματιών, αλλά η γνωριμία με τους αλγόριθμους κι η καλλιέργεια μιας πληροφορικής “κουλτούρας”, ο πληροφορικός αλφαριθμητισμός. Αντίθετα από τους ισχυρισμούς που κατά πλειοψηφία διατυπώνονται δημόσια, η δική μας θέση είναι ότι ο Πληροφορικός αλφαριθμητισμός περνάει υποχρεωτικά από τους αλγόριθμους και τον προγραμματισμό κι όχι από την απλή χρήση των ΗΥ - όπως ακριβώς οι μαθητές στο σχολείο δεν μαθαίνουν τον άψογο χειρισμό σύνθετων ηλεκτρονικών συσκευών (π.χ. της τηλεόρασης), ούτε πρακτικούς κανόνες για την επιδιόρθωσή τους αλλά τις βασικές αρχές της Φυσικής που διέπουν τη λειτουργία τους.

Στο πλαίσιο λοιπόν μιας εισαγωγής στον προγραμματισμό, το (διανοητικό) περιβάλλον του χάρτινου υπολογιστή αποτελεί ένα μέσον το οποίο επιτρέπει την παρουσίαση μερικών στοιχείων από την αλγοριθμική με σχετική ευκολία - εννοούμε με αυτό ότι η σχετική (διανοητική) πολυπλοκότητα του χάρτινου υπολογιστή εύκολα μπορεί να ελεγχθεί, σε αντίθεση με άλλα περιβάλλοντα στα οποία η πολυπλοκότητα του ίδιου του συστήματος προστίθεται στην πολυπλοκότητα των προβλημάτων τα οποία καλούνται να επιλύσουν οι μαθητές.

Το σημαντικότερο ίσως πρόβλημα που αντιμετωπίζουν οι αρχάριοι προγραμματιστές, είναι η “τυπική” συμπεριφορά του πληροφορικού συστήματος. Αντίθετα από τα φαινόμενα, οι ΗΥ υπακούουν κατά τρόπο αυστηρά καθορισμένο σε ένα σύνολο αυστηρά καθορισμένων εντολών. Οι μαθητές αντίθετα, πολλές φορές θεωρούν ότι η επικοινωνία τους με τον ΗΥ παρουσιάζει κοινά στοιχεία με την ανθρώπινη επικοινωνία - που πραγματοποιείται πάντοτε μέσα σε κάποιο καθορισμένο πλαίσιο. Συχνά λοιπόν η ανθρώπινη επικοινωνία είναι (από τυπική άποψη) ελλιπής, αλληγορική, συμβολική, συν-προσδιορίζεται από χειρονομίες, εκφράσεις κλπ, ενώ οι εντολές προς έναν ΗΥ δέχονται μία και μόνη ερμηνεία. Ο διδάσκων λοιπόν θα πρέπει να επιμένει στον χάρτινο υπολογιστή - δείχνοντας με συστηματικό τρόπο ότι ο χάρτινος υπολογιστής αποτελεί ένα “τυπικό περιβάλλον”, ένα περιβάλλον στο οποίο ισχύουν προκαθορισμένοι κανόνες κι η επίλυση οιαδήποτε προβλήματος είναι εφικτή μόνο όταν μπορεί να εκφραστεί στο πλαίσιο του χάρτινου υπολογιστή. Οι προδιαγραφές των αλγορίθμων, η κατασκευή και περιγραφή τους, η επαλήθευση ή η διάψευσή τους, η εν γένει επικύρωση τους, όλα αυτά πρέπει να υλοποιούνται μέσα στο περιβάλλον του ΗΥ και μόνο μέσα σε αυτό - εξηγώντας κάθε φορά ότι πρόκειται για ένα “τυπικό” παιχνίδι του οποίου τους κανόνες οφείλει ο προγραμματιστής να σεβαστεί απόλυτα, όπως ο σκακιστής σέβεται κατά τρόπο απόλυτο τους κανόνες του παιχνιδιού. Στην ερώτηση για παράδειγμα “τι θα

γίνονταν αν...”, ο διδάσκων πρέπει συστηματικά να προσομοιώνει την εκτέλεση του (προτεινόμενου, υποθετικού) αντίστοιχου τμήματος - αν και όχι πάντα με απόλυτη λεπτομέρεια. Η σχετική εμπειρία δείχνει ότι η εξοικείωση με τον χάρτινο υπολογιστή επιτρέπει στους μαθητές την οργάνωση των διαδικασιών επίλυσης προβλημάτων σε ένα επίπεδο υψηλότερο από εκείνο της “assembly”, δηλαδή με όρους αποκλειστικά εξαρτώμενους από τον ίδιο το χάρτινο υπολογιστή.

Ανάλογα με το επίπεδο των σπουδαστών, το περιβάλλον του χάρτινου υπολογιστή επιτρέπει και την οργάνωση στοιχειωδών αποδείξεων για την ορθότητα των προτεινόμενων αλγορίθμων κι ακόμη στοιχεία ανάλυσης της πολυπλοκότητας των αλγορίθμων.

Στο περιβάλλον του χάρτινου ΗΥ οι αλγόριθμοι μπορούν να εκφράζονται με διαγραμματικό τρόπο (οι διάφορες παραλλαγές των γνωστών διαγραμμάτων ροής) - ο διδάσκων μπορεί να επιλέξει εκείνα τα εικονικά σύμβολα που αντιστοιχούν στις επιλογές του (για παράδειγμα χρήση ή όχι του goto κλπ). Ωστόσο η χρήση του χάρτινου ΗΥ πρέπει να συνοδεύεται και από μια γλώσσα προγραμματισμού, προκειμένου οι σπουδαστές να αποκτήσουν μια σαφή εικόνα των διαφορών ανάμεσα σε μια προσομοιούμενη εκτέλεση και την πραγματική - και φυσικά να μπορούν να εκτελούν τους διάφορους αλγόριθμους που κατασκευάζουν. Οι συνήθεις γλώσσες επιτακτικού προγραμματισμού (Basic, Pascal) που χρησιμοποιούνται στην ελληνική δευτεροβάθμια εκπαίδευση είναι συμβατές με τον χάρτινο υπολογιστή.

Εξάλλου η επιλογή αυτή επιτρέπει στο διδάσκοντα να εισάγει σταδιακά τις διάφορες εντολές και έννοιες της επιλεγμένης γλώσσας προγραμματισμού όταν η επίλυση του αντίστοιχου προβλήματος το απαιτεί - κι όχι κατά τη συνήθη τακτική των εγχειριδίων να εισάγει τις εντολές όλες μαζί, συστηματικά, χωρίς αναφορά στο λόγο για τον οποίο οι εντολές αυτές επινοήθηκαν και χρησιμοποιούνται, παρουσιάζοντας δηλαδή τη σύνταξη και σημασία των εντολών (syntax, semantics), αλλά όχι και τα πραγματιστικά τους στοιχεία (pragmatics). Κατά κάποιο τρόπο η επιδίωξη του διδάσκοντος πρέπει να είναι αντίστροφη. Παρόλο που η πραγματική εκτέλεση των προγραμμάτων στον ΗΥ είναι θελκτική, είναι ωστόσο προτιμότερο να μην επιτρέψει τη χρήση ΗΥ παρά μόνο ύστερα από κάποιο εύλογο χρονικό διάστημα (2-3 μαθήματα), όταν δηλαδή οι σπουδαστές θα είναι εξοικειωμένοι τόσο με την *τεχνική* του χάρτινου υπολογιστή, όσο και με τον *διδακτικό του ρόλο* - πως δηλαδή κάθε πρόταση ή ιδέα, κάθε καινούρια λύση στα προβλήματα “εκτελείται” με λεπτομέρεια στο χάρτινο ΗΥ προκειμένου να διαπιστωθεί αν λειτουργεί έτσι όπως προβλέπεται. Είναι επίσης προτιμότερο στα αρχικά μαθήματα να μη γίνει εισαγωγή σε όλες τις λεπτομέρειες και περιπτώσεις που καλύπτει η επιλεγμένη γλώσσα προγραμματισμού - για παράδειγμα όλα τα format εκτυπώσεων. Η πρόσθετη (διανοητική) πολυπλοκότητα των κανόνων που διέπουν, για παράδειγμα, τις εκτυπώσεις ή τη συμβατότητα των τύπων μπορεί να εκτρέψει τη ροή της διδασκαλίας: όπως συχνά συμβαίνει, οι σπουδαστές μπορούν να εμπλακούν στις λεπτομέρειες των εκτυπώσεων και να χάσουν το βασικό στόχο που είναι - για παράδειγμα - η παραγωγή των ακεραίων από 1 έως N και των τετραγώνων τους. Εξάλλου σύμφωνα και με όσα πρωτύτερα αναπτύξαμε, οι λεπτομέρειες της γλώσσας ή τα ιδιοσυγκρασιακά της στοιχεία αποτελούν κατά κανόνα εργαλεία, ανταποκρίνονται δηλαδή σε μια προβληματική, την οποία ο διδάσκων πρέπει να σεβαστεί, προκειμένου αυτές να αποκτήσουν νόημα.

Παραθέτουμε ένα παράδειγμα. Μια από τις ισχυρές αυθόρμητες αντιλήψεις των νεαρών μαθητών συνδέεται με τις εμφανίσεις των αποτελεσμάτων: οι νεαροί αρχάριοι προγραμματιστές θεωρούν ότι οι εμφανίσεις αποτελεσμάτων στην οθόνη ακολουθούν τους έμμεσους κανόνες γραφής που τηρούνται από τους ανθρώπους. Αν για παράδειγμα κατά την εκτέλεση ενός προγράμματος δεν εμφανιστούν αποτελέσματα για αρκετή ώρα, η αυθόρμητη αντίδρασή τους είναι ότι “ο ΗΥ χάλασε” ή “κόλλησε” κι όχι ότι οι ίδιοι δεν έβαλαν κάποια εντολή εκτύπωσης στο πρόγραμμά τους - γιατί θεωρούν ότι από τη στιγμή που υπάρχουν αποτελέσματα, θα πρέπει αυτά, αυτομάτως, να εμφανιστούν. Η εντολή $X:=8+5$ της Pascal θα προκαλέσει την αποθήκευση του αριθμού 13 στη μεταβλητή X - καμία εμφάνιση αποτελεσμάτων - ενώ στην ανθρώπινη επικοινωνία η εντολή “υπολόγισε το $8+5$ ”, ή “πόσο κάνουν $8+5$ ”, σημαίνει έμμεσα και την ανακοίνωση του αποτελέσματος “13” (προφορικά ή γραπτά). Επίσης η εμφάνιση των αποτελεσμάτων στην οθόνη πρέπει να ακολουθεί - αυτόματα - τις τρέχουσες συμβάσεις: για παράδειγμα τα στοιχεία ενός πίνακα $M \times N$ πρέπει να εμφανιστούν σε “ορθογώνια” διάταξη. Όταν τα αποτελέσματα εμφανίζονται σε μια στήλη, η έκπληξη τους είναι μεγάλη - πολλές φορές θεωρούν ότι ο ΗΥ “έχει κάνει λάθος” γιατί η τοπολογική διάταξη των στοιχείων δεν ανταποκρίνεται στο αναμενόμενο αποτέλεσμα. Είναι ακριβώς στην περίπτωση αυτή που οι διάφοροι κανόνες για την εμφάνιση των αποτελεσμάτων αποκτούν νόημα κι επομένως ο διδάσκων μπορεί να τους παρουσιάσει.

Μέχρις ενός σημείου ο χάρτινος υπολογιστής επιτρέπει μια ερμηνεία των χαρακτηριστικών της γλώσσας προγραμματισμού, όπως οι τύποι, οι προσδιορισμοί των μεταβλητών (π.χ. στην Pascal), οι δείκτες, το πέρασμα των παραμέτρων, η αναδρομή.

Τέλος, στο πλαίσιο του χάρτινου υπολογιστή, και πάντα ανάλογα με το επίπεδο του κοινού και τους στόχους του μαθήματος, είναι δυνατή μια στοιχειώδης ανάλυση των αλγορίθμων, από την πλευρά της αποτελεσματικότητας τους: για παράδειγμα είναι εύκολο να δειχθεί ότι η πολυπλοκότητα δυο εμφωλευμένων βρόχων που εκτελούνται M και N φορές αντίστοιχα, είναι $O(M*N)$ κλπ.

Βασική έννοια είναι η έννοια του *αλγορίθμου*: είναι μια διαδικασία η οποία σε πεπερασμένου πλήθους βήματα επιλύει ένα πρόβλημα. Τι θα πει “βήματα”; Κάθε βήμα είναι μια πράξη την οποία ξέρει να εκτελέσει ο ΗΥ. Ο διδάσκων μπορεί να παρουσιάσει μερικούς αλγόριθμους για να εξηγήσει ορισμένες βασικές έννοιες. Συνηθίζεται πολλές φορές να δίνονται παραδείγματα αλγορίθμων από τις συνταγές της μαγειρικής - τα οποία είναι εύκολο να κατανοηθούν. Θεωρούμε ωστόσο όχι και πολύ πετυχημένη την εκλογή τους - γιατί υπάρχει ο κίνδυνος να δοθεί μια λανθασμένη εντύπωση σχετικά με την έννοια του αλγορίθμου. Αντίθετα θα προτεινάμε ένα καθαρό μαθηματικό παράδειγμα - ας πούμε τον υπολογισμό των ριζών μιας δευτεροβάθμιας εξίσωσης.

Ας υποθέσουμε ότι ένας μαθητής παίζει το ρόλο του ΗΥ. Ένας αλγόριθμος γράφεται στο χαρτί και δίνεται στο μαθητή. Το χαρτί περιέχει τις εξής εντολές:

1. Ζήτησε (από τον διδάσκοντα) έναν αριθμό. Τον αριθμό αυτό ονόμασέ τον A.
2. Ζήτησε (από τον διδάσκοντα) έναν αριθμό. Τον αριθμό αυτό ονόμασέ τον B.
3. Ζήτησε (από τον διδάσκοντα) έναν αριθμό. Τον αριθμό αυτό ονόμασέ τον C.

4. Υπολόγισε το $B*B$ και το αποτέλεσμα ονόμασε το D.
5. Υπολόγισε το $4*A*C$ κι ονόμασε το E
6. Υπολόγισε το D-E κι ονόμασε το K
7. Αν το K είναι αρνητικό τότε σημείωσε στον πίνακα “δεν υπάρχουν πραγμ. λύσεις”. Πήγαινε μετά στην εντολή 12
8. Υπολόγισε την τετραγωνική ρίζα του K κι ονόμασε την L
9. Υπολόγισε την ποσότητα $(-B+L)/(2*A)$ κι ονόμασε την X1
10. Υπολόγισε την ποσότητα $(-B-L)/(2*A)$ κι ονόμασε την X2
11. Γράψε τα X1, X2 στον πίνακα
12. Τέλος

Είναι εύκολο να δει κανείς ότι η εκτέλεση των εντολών αυτών οδηγεί στο επιθυμητό αποτέλεσμα - τη λύση μιας δευτεροβάθμιας εξίσωσης, όταν δοθούν οι συντελεστές.

Δυο παρατηρήσεις επιβάλλονται: ο μαθητής που παίζει το ρόλο του ΗΥ δεν χρειάζεται να “καταλαβαίνει” τι κάνει: πρέπει απλώς να ξέρει να εκτελεί τις εντολές που είναι γραμμένες στο χαρτί. Το αποτέλεσμα θα είναι ορθό ανεξάρτητα από τον “ΗΥ”. Για να πειστούν οι μαθητές, μπορεί ο διδάσκων να τους ζητήσει να ακολουθήσουν τον εξής αλγόριθμο (κατά προτίμηση από τους μαθητές που διαθέτουν μια αριθμομηχανή τσέπης):

Ζήτησε (από τον διδάσκοντα) έναν αριθμό κι ονόμασε τον A

Διάλεξε έναν φυσικό αριθμό ανάμεσα στο 1 και το 1000 κι ονόμασε τον X

Υπολόγισε την ποσότητα $X+$

Στο τέλος οι μαθητές έχουν - κατά προσέγγιση - υπολογίσει το ίδιο αποτέλεσμα - χωρίς ωστόσο να ξέρουν τι. Ο διδάσκων μπορεί να τους εξηγήσει ότι υπολόγισαν την τετραγωνική ρίζα του A με προσέγγιση εκατοστού - έστω κι αν δεν ήξεραν τι έκαναν.

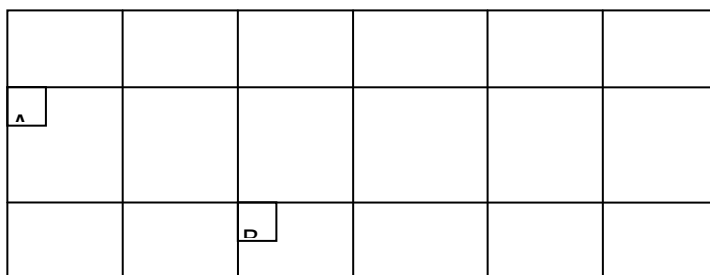
Σημείωση: Ανάλογα με το διαθέσιμο χρόνο και το επίπεδο του ακροατηρίου, ο διδάσκων θα μπορούσε να οργανώσει την τάξη ως ένα είδος ΗΥ: ένας μαθητής θα προσομοιώνει το bus, άλλος την κεντρική μονάδα επεξεργασίας, άλλοι τις μνήμες κλπ. Ο “ΗΥ” αυτός θα μπορούσε να εκτελέσει τους παραπάνω αλγόριθμους και να παραγάγει τα επιθυμητά αποτελέσματα.

Μια δεύτερη παρατήρηση που επιβάλλεται είναι σχετική με τις δυνατότητες του ΗΥ: ποιες πράξεις μπορεί να κάνει; Ο χάρτινος ΗΥ που περιγράφεται παρακάτω, καθορίζει με αρκετή σαφήνεια τις πράξεις που είναι επιτρεπτές για την περιγραφή ενός αλγόριθμου - έτσι ώστε αυτός εύκολα να γίνεται πρόγραμμα.

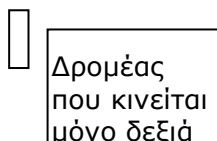
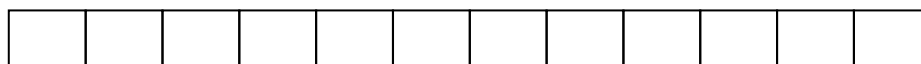
Ένας ΗΥ αποτελείται, σύμφωνα με το μοντέλο του χάρτινου ΗΥ, από τα εξής μέρη:

μια σειρά από κυψέλες οι οποίες ονομάζονται “μνήμες” και κάθε μια από αυτές μπορεί να αποθηκεύσει μια στοιχειώδη πληροφορία - έναν αριθμό για παράδειγμα. Σε κατοπινό στάδιο μπορεί να εξηγηθεί ότι οι μνήμες δεν είναι ισοδύναμες κι η κάθε μια μπορεί να αποθηκεύσει την τιμή μιας μεταβλητής ενός ορισμένου τύπου (έναν ακέραιο, έναν πραγματικό, μια συμβολοσειρά κλπ. Επίσης μπορεί να εξηγηθεί το μέγεθος της πραγματικής μνήμης που απαιτείται για την αποθήκευση μιας τιμής - 2,4 6 ή περισσότερα bytes). Για την αρχή είναι αρκετό να υποθεθεί ότι κάθε μια κυψέλη μπορεί να αποθηκεύσει μια ακεραία τιμή. Κάθε μνήμη έχει ένα όνομα - που το επιλέγει ο προγραμματιστής - και πιθανόν ένα περιεχόμενο. Η επιλογή της μνήμης είναι τυχαία. Δηλαδή αν ο προγραμματιστής ονομάσει

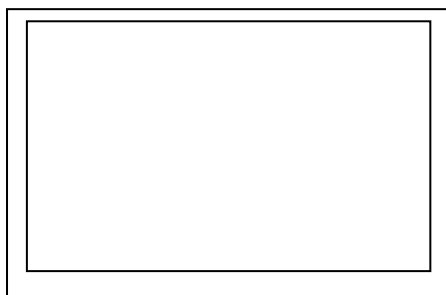
μια μνήμη A και αποθηκεύσει μέσα σε αυτήν τον αριθμό, δεν γνωρίζει ποια είναι η (φυσική) μνήμη που ονομάστηκε A - αλλά στη συνέχεια κάθε φορά που θα αναφέρεται στη μνήμη A θα εννοείται η ίδια (φυσική) μνήμη.



- μια πρακτικά ατέρμονη ταινία με κυψέλες οι οποίες λειτουργούν όπως οι μνήμες - κάθε μια χωράει μόνο μια στοιχειώδη πληροφορία κλπ. Ένας δρομέας ξεκινάει από “αριστερά” και κινείται προς τα “δεξιά” δείχνοντας κάθε φορά και μια διαφορετική κυψέλη. Ο δρομέας κινείται μόνο δεξιά.



- μια “αόρατη” μονάδα επεξεργασίας που εκτελεί πράξεις
- μια “οθόνη” στην οποία κατευθύνονται όλες οι εκτυπώσεις.



Η εξαιρετικά απλουστευμένη αυτή αναπαράσταση του ΗΥ είναι σχετικά απλή στη λειτουργία της, αλλά προσφέρει μεγάλες δυνατότητες υπολογισμού.

Ο προγραμματιστής μπορεί να επιδράσει (δηλαδή να προγραμματίσει) στο χάρτινο ΗΥ με τις εντολές που περιγράφονται παρακάτω. Κάθε εντολή μπορεί να παρασταθεί διαγραμματικά ή με κείμενο (γραπτές εντολές):

ΔΙΑΒΑΣΜΑ: (ατυχής επιλογή ενός όρου για να ονοματιστεί μια έννοια που έτσι κι αλλιώς προκαλεί σύγχυση). Το διάβασμα έχει νόημα μόνο σε συνδυασμό με κάποια μνήμη:

Διαγραμματική αναπαράσταση



Διατύπωση εντολών με κείμενο

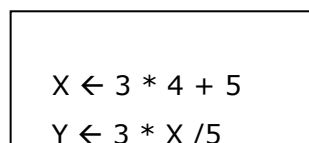
Διάβασε MIN

Η εντολή αυτή σημαίνει “τοποθέτησε το επόμενο δεδομένο στη μνήμη MIN”.

Σε κάθε διάβασμα συμβαίνουν τα εξής. Ο δρομέας της ατέρμονης ταινίας προχωράει μια θέση δεξιά. Αν η κυψέλη την οποία “αντικρύζει” είναι κενή, υπάρχει λάθος: στην οθόνη εμφανίζεται ένα μήνυμα λάθους κι η εκτέλεση του αλγορίθμου σταματάει. ΑΝ η αντίστοιχη κυψέλη δεν είναι κενή, ο ΗΥ ψάχνει να βρει μια μνήμη με την ονομασία MIN. ΑΝ δεν τη βρει, τότε διαλέγει μια από τις διαθέσιμες μνήμες, την ονομάζει MIN και μέσα σε αυτήν τοποθετεί τον αριθμό που “αντικρύζει” ο δρομέας.

ΕΚΧΩΡΗΣΗ ΚΙ ΕΚΤΕΛΕΣΗ ΠΡΑΞΕΩΝ: ο ΗΥ εκτελεί διάφορες πράξεις κι εκχωρεί (δηλαδή αποθηκεύει) το αποτέλεσμα σε μια μνήμη. Στο παρακάτω παράδειγμα:

Διαγραμματική αναπαράσταση



Διατύπωση εντολών με κείμενο

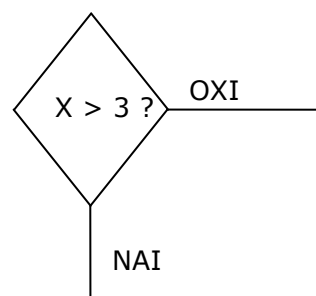
X ← 3*4+5

Y ← 3 * X /5

Το αποτέλεσμα των δυο εκχωρήσεων είναι να εκχωρηθεί ο αριθμός 17 στη μνήμη X κι αριθμός 10,2 (51/5) στη μνήμη Y.

ΣΥΓΚΡΙΣΗ ΚΑΙ ΔΙΑΚΛΑΔΩΣΗ:

Διαγραμματική αναπαράσταση



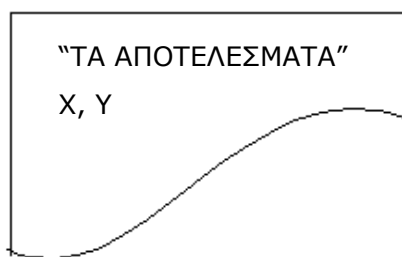
Διατύπωση εντολών με κείμενο

Αν $X > 3$ τότε αλλιώς

Ο αριθμός X (ακριβέστερα το περιεχόμενο της μνήμης $X0$ συγκρίνεται με το 3. Αν είναι μεγαλύτερο τότε ο αλγόριθμος συνεχίζει από τον κλάδο “ΝΑΙ”, αλλιώς από τον κλάδο “ΟΧΙ”. Οι ερωτήσεις αυτές (που επιδέχονται σα μοναδική απάντηση ΝΑΙ / ΟΧΙ) είναι οι μοναδικές που επιτρέπονται.

ΕΜΦΑΝΙΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ:

Διαγραμματική αναπαράσταση



Διατύπωση εντολών με κείμενο

Τύπωσε “ΤΑ ΑΠΟΤΕΛΕΣΜΑΤΑ”, X, Y

Ότι είναι ανάμεσα σε εισαγωγικά εμφανίζεται στην οθόνη ακριβώς όπως είναι. Τα υπόλοιπα (στο παραπάνω παράδειγμα τα X και Y) πρέπει να είναι υποχρεωτικά ονόματα μεταβλητών, οπότε στην οθόνη εμφανίζεται το περιεχόμενο της αντίστοιχης μνήμης.

Κάθε διάγραμμα έχει μια αρχή κι ένα τέλος και η πορεία του αλγορίθμου στη διαγραμματική αναπαράσταση επισημαίνεται με βέλη που οδηγούν από το ένα σύμβολο στο άλλο, ενώ στη διατύπωση εντολών με κείμενο οι εντολές εκτελούνται με τη σειρά που διαβάζουμε, ενώ κάθε “μπλοκ εντολών” εγκλείεται μέσα σε άγκιστρα {}.

Στα πλαίσια αυτά, ονομάζεται *αλγόριθμος* μια διαδικασία που περιγράφεται πλήρως με τις εντολές του χάρτινου υπολογιστή κι επιλύει ένα πρόβλημα σε πεπερασμένο πλήθος βημάτων (απαραίτητο για να γίνει διάκριση σε σχέση με τα Μαθηματικά). Ένας αλγόριθμος μπορεί να εκφραστεί με απλές προτάσεις, με διαγράμματα, με τη βοήθεια μιας γλώσσας προγραμματισμού κλπ. Ένας αλγόριθμος που είναι εκφρασμένος σε μια γλώσσα προγραμματισμού ονομάζεται *πρόγραμμα*. Όταν οι γνώσεις των μαθητών θα είναι πιο βαθιές - κι αν βέβαια το απαιτεί ο στόχος του μαθήματος - ο διδάσκων μπορεί να τροποποιήσει τους ορισμούς αυτούς.

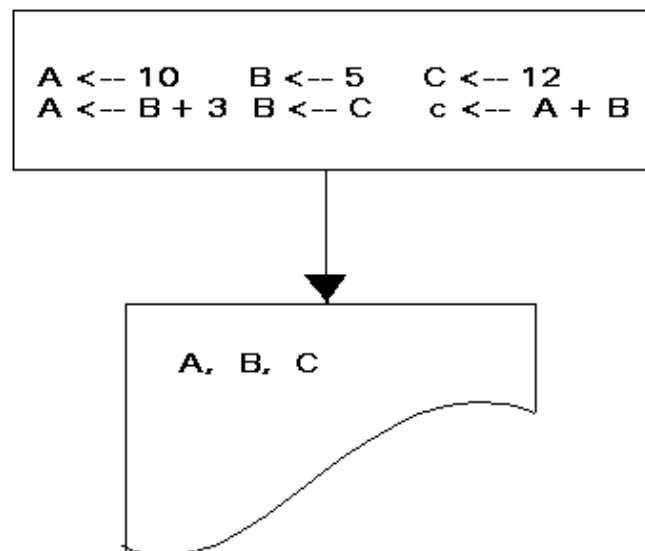
Συχνά η φαινομενική “φτώχεια” του μοντέλου προκαλεί κάποια αμφιβολία στους μαθητές: είναι δυνατόν να γίνονται όλα με τις απλές αυτές εντολές; Βέβαια το μοντέλο αυτό είναι πράγματι ελλιπές: απουσιάζουν για παράδειγμα οι δυνατότητες για γραφικά (η οθόνη, έτσι όπως περιγράφεται επιτρέπει την εκτύπωση μόνο χαρακτήρων, η δυνατότητα επικοινωνίας

με το σύστημα, με άλλα προγράμματα κλπ. Ο χάρτινος υπολογιστής περιλαμβάνει ένα μέρος λοιπόν των δυνατοτήτων των ΗΥ - και τούτο προκειμένου να μειώσει τη σχετική (διανοητική) πολυπλοκότητα ενός πλήρους σχήματος, μιας πλήρους δηλαδή παρουσίασης των δυνατοτήτων των ΗΥ. Ο διδάσκων μπορεί λοιπόν να εξηγήσει στους μαθητές ότι ένα μεγάλο τμήμα των εργασιών αντιστοιχεί στις δυνατότητες του χάρτινου υπολογιστή.

Προβλήματα για μια εισαγωγή στον προγραμματισμό

1. Τα πρώτα προβλήματα δεν είναι παρά βήματα εξοικείωσης με τον χάρτινο υπολογιστή. Ο διδάσκων θα πρέπει να επιμείνει στη λεπτομερή εκτέλεσή τους στον πίνακα (στο χάρτινο υπολογιστή). Δίνεται παρακάτω ένα τέτοιο παράδειγμα.

Ζητείται να προσδιοριστεί το αποτέλεσμα των παρακάτω πράξεων:



2. Να προσδιοριστεί το αποτέλεσμα των παρακάτω εντολών:

Διάβασε έναν αριθμό στη μνήμη A

Τοποθέτησε 5 στη μνήμη B

Πρόσθεσε τα A και B και το αποτέλεσμα βάλτο στη μνήμη A

Πολλαπλασίασε τη μνήμη A επί 2 και το αποτέλεσμα βάλτο στη μνήμη A

Τύπωσε το περιεχόμενο της μνήμης A

3. Να κατασκευαστεί ένας αλγόριθμος ο οποίος να υπολογίζει το άθροισμα των 3,6,8 και να το εκτυπώνει (δηλαδή να το εμφανίζει στην οθόνη). Στη συνέχεια να κατασκευαστεί ένας αλγόριθμος ο οποίος να κάνει το ίδιο για τους αριθμούς 31, 500, 700. Να κατασκευαστεί τέλος ένας αλγόριθμος ο οποίος να προσθέτει τρεις αριθμούς.

Στην άσκηση αυτή θα πρέπει να τονιστεί η ιδιότητα των αλγορίθμων να επιλύουν κλάσεις προβλημάτων - κι όχι μεμονωμένα προβλήματα. Η τρίτη παραλλαγή στο πρόβλημα 3 ζητάει μια μέθοδο υπολογισμού του αθροίσματος τριών αριθμών που μας είναι για την ώρα άγνωστοι. Θα πρέπει να τονιστεί ότι αυτό ισχύει γενικά για τους αλγόριθμους - ουσιαστικά αποτελούν περιγραφές γενικών μεθόδων επίλυσης προβλημάτων.

4. Με ποιο τρόπο θα αντιμετωπίσουμε δυο τιμές; Αν για παράδειγμα η μνήμη A περιέχει τον αριθμό 8 και η B τον 5, με ποιο τρόπο θα αντιμετωπίσουμε τα περιεχόμενα τους;

Κατά κανόνα ένα τμήμα των σπουδαστών προτείνει αμέσως μια διπλή εκχώρηση:

$A \leftarrow B, B \leftarrow A$. Η προσφυγή στο χάρτινο υπολογιστή δείχνει αμέσως ότι το ζεύγος αυτό των εντολών δεν αντιμετωπίζει τις τιμές των A και B.

Μερικοί σπουδαστές προτείνουν κατευθείαν τις εκχωρήσεις $A \leftarrow 5$ και $B \leftarrow 8$. Και πάλι χρησιμοποιώντας το χάρτινο υπολογιστή είναι εύκολο να διαπιστωθεί ότι αυτό το ζεύγος εντολών λειτουργεί σωστά για τους συγκεκριμένους αριθμούς, αλλά δεν συνιστά γενική μέθοδο. Αν τα περιεχόμενα των μνημών A και B είναι άλλα, προφανώς το αποτέλεσμα θα είναι λάθος.

Στο σημείο αυτό εύκολα διαπιστώνεται η αναγκαιότητα χρήσης και μια τρίτης μεταβλητής, έστω C, για την προσωρινή αποθήκευση μιας τιμής.

Ανάλογα με το επίπεδο των σπουδαστών ο διδάσκων μπορεί να προτείνει (για αριθμητικά δεδομένα) και τις εκχωρήσεις

$$A \leftarrow B + A, B \leftarrow A - B, A \leftarrow A - B$$

Στην περίπτωση αυτή δεν χρειάζεται τρίτη μεταβλητή C.

Μια ή δυο δοκιμές πείθουν τους μαθητές ότι αυτές οι εντολές λειτουργούν σωστά. Η ορθότητα τους μπορεί να αποδειχθεί με πολύ απλή άλγεβρα.

Αν η μνήμη έχει αρχικό περιεχόμενο περιεχόμενο χ και η B έχει αρχικό περιεχόμενο ψ , τότε η απόδειξη έχει ως εξής:

<i>Εκχωρήσεις</i>	<i>Τιμές των μεταβλητών</i>
αρχικές τιμές	$A=\chi$ και $B=\psi$
$A \leftarrow A + B$	$A=\chi+\psi$ και $B=\psi$
$B \leftarrow A - B$	$A=\chi+\psi$ και $B=A-B=(\chi+\psi)-\psi=\chi$
$A \leftarrow A - B$	$A=A-B=(\chi+\psi)-\chi=\psi$
τελικές τιμές	$A=\psi$ και $B=\chi$

5. Να κατασκευασθεί ένας αλγόριθμος ο οποίος από δύο δεδομένους αριθμούς τυπώνει τον μικρότερο. Στη συνέχεια να κατασκευαστεί ένας αλγόριθμος ο οποίος από 3 αριθμούς θα εκτυπώνει τον μεγαλύτερο. Τέλος να κατασκευαστεί ένας αλγόριθμος ο οποίος θα εκτυπώνει τρεις αριθμούς A,B,C σε φθίνουσα σειρά.

Το πρόβλημα αποσκοπεί στην εξοικείωση με τις εντολές επιλογών. Θα πρέπει να τονιστεί ότι κάθε if πρέπει να δέχεται ως απάντηση ένα “ΝΑΙ” ή ένα “ΟΧΙ” - γεγονός που αυξάνει τον αριθμό των απαιτούμενων ερωτήσεων - αλλά αυτό αποτελεί μέρος των τυπικών κανόνων λειτουργίας του χάρτινου υπολογιστή κι είναι αναπόφευκτο. Για άλλη μια φορά θα πρέπει να τονιστεί το πόσο διαφορετική είναι η λύση ακόμη κι ενός τετριμμένου προβλήματος με χαρτί-μολύβι και προγραμματιστικά.

6. Υπολογισμός ενός μαθηματικού τύπου: για παράδειγμα δεδομένης της ακτίνας ενός κύκλου να υπολογιστεί το μήκος της περιφέρειας του και το εμβαδόν του. Το πρόβλημα αποτελεί μια ευκαιρία για να τονιστεί ο κατ' ανάγκη προσεγγιστικός χαρακτήρας των αποτελεσμάτων. Η μαθηματική γραφή

$$L = 2 \pi r$$

αποτελεί μια έκφραση του ακριβούς μήκους της περιφέρειας συναρτήσει της ακτίνας. Η αριθμητική έκφραση του αποτελέσματος μπορεί - θεωρητικά - να προσεγγίσει το πραγματικό αποτέλεσμα, αρκεί να δοθεί το r και το π με την απαιτούμενη ακρίβεια (σήμερα είναι γνωστά τα πρώτα 10.000.000 ψηφία του π).

Αντίθετα η εκχώρηση

$$L := 2 * \pi * R$$

δε μπορεί παρά να παράγει ακρίβεια ορισμένων μόνο ψηφίων (ανάλογα με τις δυνατότητες της γλώσσας προγραμματισμού).

Ποια είναι όμως η ακρίβεια της διαγραμματικής αλγοριθμικής γλώσσας; Εκτός κι αν ρητά αναφέρεται θα θεωρείται ότι είναι εκείνη των μαθηματικών. Θα πρέπει ίσως να τονιστεί ότι στο πλαίσιο της εισαγωγής στους αλγορίθμους δε λαμβάνονται υπόψη, αρχικά τουλάχιστον, τα προβλήματα αυτά. Η εισαγωγή όμως και χρήση μιας γλώσσας προγραμματισμού θα προκαλέσει αναπόφευκτα την αντιμετώπισή τους.

Σημείωση: Θα πρέπει ίσως να τονιστεί στο σημείο αυτό μια ιδιαίτερη δυσκολία που υπάρχει σχετικά με δυο συγγενή προβλήματα:

πρόβλημα 1^ο: η αναπαράσταση της ταινίας δεδομένων στην Pascal. Η ταινία με τα δεδομένα στη Basic αντιστοιχεί στο ζεύγος DATA - READ, αλλά στην Pascal δεν αντιστοιχεί σε τίποτε. Ο διδάσκων θα πρέπει να εξηγήσει την αντιστοιχία ανάμεσα στην ταινία και την εντολή

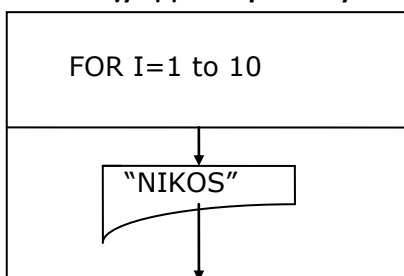
Read/ReadIn της Pascal. Αυτό οδηγεί κατευθείαν στο επόμενο πρόβλημα

πρόβλημα 2^ο: η διαχείριση της εισόδου δεδομένων. Η εισαγωγή δεδομένων κατευθείαν από το πληκτρολόγιο αποτελεί ένα δύσκολο θέμα για τον αρχάριο προγραμματιστή. Το πρόγραμμα φαίνεται να σταματάει να λειτουργεί κι ένας δρομέας αναβοσβήνει σε ένα σημείο της οθόνης. Οι μαθητές δυσκολεύονται κατά κανόνα να ερμηνεύσουν αυτό το προσωρινό “πάγωμα” ως αποτέλεσμα της δικής τους εντολής read. Θα πρέπει να γίνουν αρκετές επαναλήψεις ώστε να μπορέσουν οι μαθητές να κατανοήσουν τη λειτουργία των εντολών εισόδου από το πληκτρολόγιο απευθείας. Όπως και παραπάνω αναφέραμε, αυτό δεν λύνει το πρόβλημα τελείως, δεδομένου ότι συχνά οι αρχάριοι προγραμματιστές θεωρούν απαραίτητες τις εντολές αυτές ακόμη κι όταν δεν χρειάζονται.

7. Να τυπωθεί στην οθόνη 10 φορές μια λέξη. Θα εισαχθεί η έννοια του βρόχου for... θα πρέπει να γίνει λεπτομερής παρουσίαση της λειτουργίας του: να δειχθεί ότι πραγματοποιείται ένας έλεγχος κάθε φορά με τη βοήθεια ενός (έμμεσου) *μετρητή*. Να εξηγηθεί ότι ο μετρητής δεν είναι παρά μια μεταβλητή (μνήμη) κι επομένως μπορεί να χρησιμοποιηθεί όπως οι άλλες μεταβλητές - αλλά δεν πρέπει να αποτελέσει target μιας εκχώρησης γιατί τα αποτελέσματα μπορεί να είναι καταστροφικά. Σα συνέχεια να ζητηθεί να τυπωθούν στην οθόνη οι αριθμοί από το 1 ως το 10. Στην δεύτερη αυτή άσκηση εκτυπώνεται ο ίδιος ο μετρητής. Τέλος, αν ο χρήστης εισάγει έναν (ακέραιο) αριθμό N, ο ΗΥ θα εκτυπώσει τους αριθμούς από 1 ως N.

Για άλλη μια φορά θα πρέπει να τονιστεί το γεγονός ότι τη στιγμή που γράφεται το πρόγραμμα, ο N μπορεί να μην είναι γνωστός - πρέπει απλώς να είναι γνωστός τη στιγμή που το πρόγραμμα εκτελείται.

Διαγραμματική αναπαράσταση



Διατύπωση εντολών με κείμενο

Για I=1 ως 10

Τύπωσε “ΝΙΚΟΣ”

Για I=1 ως 10

Τύπωσε I

Διάβασε N

Για I=1 ως N

Τύπωσε I

8. Να προσδιορισθεί ο ελάχιστος από 20 αριθμούς. Να υπολογιστεί ο ελάχιστος από N αριθμούς. Τρία σημεία πρέπει να επισημανθούν:

σημείο 1: η λύση που ακολουθείται είναι βέβαια τελείως διαφορετική από μια “λύση” με χαρτί και μολύβι όπου μπορεί κανείς να εντοπίσει τον ελάχιστο απλώς κοιτώντας διαδοχικά τους αριθμούς.

σημείο 2: ανάλογα με το επίπεδο των μαθητών μπορεί κανείς να σημειώσει το αναλλοίωτο σε κάθε βρόχο - το γεγονός δηλαδή ότι στην αρχή του κάθε βρόχου η μνήμη MIN περιέχει τον ελάχιστο από όσους αριθμούς εξετάστηκαν ως τότε και στο τέλος κάθε βρόχου ισχύει το ίδιο με αυξημένο το πλήθος των εξετασμένων στοιχείων κατά 1.

σημείο 3: ανάλογα με το επίπεδο των μαθητών μπορεί κανείς να πραγματοποιήσει μια στοιχειώδη ανάλυση των αναμενόμενων - κατά μέσο όρο - αλλαγών της μνήμης MIN.

Διάβασε N

Αν N=0 τότε τύπωσε “δεν υπάρχουν δεδομένα”

αλλιώς

{ Διάβασε X

MIN = X

Αν N>1 τότε { Για I=2 ως N

{ Διάβασε X

Αν X < MIN τότε MIN = X

}

}

Τύπωσε MIN

}

Σαν πρόσθετες ασκήσεις θα μπορούσαν να προταθούν ο προσδιορισμός του μεγίστου, κι ο ταυτόχρονος υπολογισμός μεγίστου κι ελαχίστου.

9. Για δεδομένο N να υπολογιστεί το άθροισμα $1+2+3+\dots+N$

Η άσκηση αυτή μοιάζει να είναι πολύπλοκη για τους αρχάριους προγραμματιστές (όπως το έχουμε επισημάνει και σε προηγούμενες παραγράφους). Θα πρέπει ο διδάσκων να δείξει με σαφήνεια το γεγονός ότι η μεταβλητή I του βρόχου είναι μια “κανονική” μεταβλητή, η οποία παίζει το ρόλο ενός μετρητή και δεν επιτρέπεται βέβαια να αλλάζουμε τις τιμές της, αλλά είναι απολύτως νόμιμο να *χρησιμοποιήσουμε* τις διαδοχικές της τιμές.

Διάβασε N

S=0

Για I=1 ως N

$S=S+I$

Τύπωσε S

10. Δίνονται N πεντάδες αριθμών. Κάθε πεντάδα αντιστοιχεί σε ένα άτομο ως εξής:
Ο πρώτος αριθμός αντιστοιχεί στον κωδικό του ατόμου, ο δεύτερος στο φύλο (0 αν είναι άνδρας και 1 αν είναι γυναίκα), ο τρίτος το βάρος του ατόμου (σε κιλά), ο τέταρτος το ύψος του σε εκατοστά κι ο τελευταίος την ηλικία του. Ζητείται το βάρος του υψηλότερου άνδρα.
Το πρόβλημα με την άσκηση αυτή είναι ότι οι μαθητές συχνά δεν διαβάζουν όλα τα δεδομένα - γιατί θεωρούν, με ένα είδος εγωκεντρισμού, ότι αυτό δεν είναι απαραίτητο αφού δεν χρειάζονται όλα τα στοιχεία (π.χ. η ηλικία του ατόμου). Η άσκηση βοηθάει λοιπόν στο να τονιστεί για άλλη μια φορά το γεγονός ότι ο τυπικός τρόπος λειτουργίας των ΗΥ δεν έχει καμία σχέση με τον ευέλικτο τρόπο συμπεριφοράς των ανθρώπινων όντων.

Διάβασε N

MAX=0

Για I=1 ως N

{ Διάβασε Cod, F, B, Y, H

 Αν F=0 τότε

 { Αν B>MAX τότε

 { MAX=B

 LOC=Cod

 }

 }

}

While και Repeat. Η διάκριση ανάμεσα στα δυο είδη βρόχων φαίνεται ότι παρουσιάζει προβλήματα ακόμη και για μη αρχάριους προγραμματιστές. Ίσως η καταλληλότερη διδακτική στρατηγική είναι τα προβλήματα στα οποία η επιλογή ανάμεσα στα δυο είδη βρόχων είναι ουσιαστική - δηλαδή ο κατάλληλος βρόχος διευκολύνει την επίλυση του αντίστοιχου προβλήματος. Δύο τυπικά τέτοια παραδείγματα είναι τα εξής:

11. Σε μια πόλη κυκλοφορούν 125 000 αυτοκίνητα. Η κυκλοφορία των αυτοκινήτων αυξάνει με ένα μέσο ετήσιο ρυθμό 14%. Να προσδιορισθεί μετά από πόσα χρόνια τα αυτοκίνητα θα είναι πάνω από 180 000.

12. Δίνεται μια σειρά από αριθμούς που παριστάνουν τα ύψη μαθητών. Το πλήθος των αριθμών είναι άγνωστο, αλλά είναι γνωστό ότι ο τελευταίος αριθμός της σειράς είναι 0 (ο οποίος προφανώς δεν αποτελεί ύψος κι άρα δεν πρέπει να προσμετρηθεί. Να προσδιοριστεί το μέσο ύψος των μαθητών. Η τεχνική αυτή - του τελευταίου στοιχείου το οποίο απλώς

επισημαίνει το τέλος των δεδομένων - είναι πού συνηθισμένη - ακόμη και τα αρχεία έχουν ένα ειδικό σύμβολο EOF (end of file) για την επισήμανση αυτή.

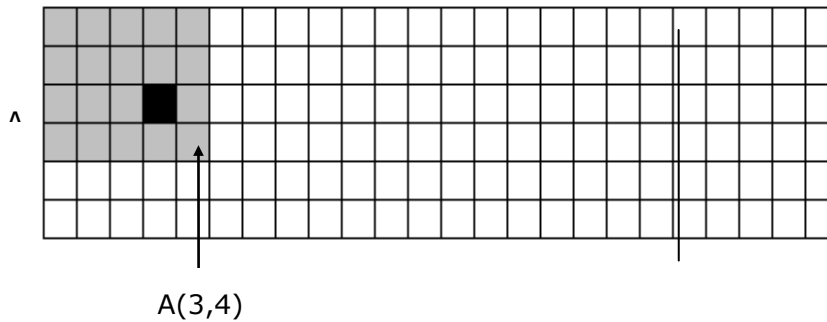
N=125 000	<u>Διάβασε</u> X
Limit =180 000	<u>Εφόσον</u> X<>0
επανάλαβε	{ S=S+X
N= ακέραιο μέρος (N*1,14)	MET=MET+1
ώσπου N>180 000	}
	MO=S/MET
	<u>Τύπωσε</u> MO

13. Να κατασκευαστούν πίνακες πολλαπλασιασμού (“προπαιδείας”), από το 1 ως το 10.
Η άσκηση αποσκοπεί στη χρήση διπλών (εμφωλευμένων βρόχων)

```
Για I=1 ως 10
  Για I=1 ως 10
    { Product = I * J
      Τύπωσε I, “x”, J, “=”, Product
    }
```

Η άσκηση προσφέρει την ευκαιρία για μια στοιχειώδη προσέγγιση στο θέμα της πολυπλοκότητας κι αποτελεσματικότητας των αλγορίθμων. Είναι προφανές ότι οι εντολές στο εσωτερικό των βρόχων θα εκτελεστούν 100 φορές = 10 x 10. Με μια γενικότερη θεώρηση θα λέγαμε ότι θα εκτελεστούν M x N φορές. Μπορεί επίσης να εξηγηθεί ότι πιθανές άλλες εντολές του κόμματος έχουν “μικρότερη πολυπλοκότητα” κι άρα μπορούν να παραλειφθούν (πολυπλοκότητα O(M x N) αν ο ένας βρόχος εκτελεστεί M φορές κι ο άλλος N).

14. Ασκήσεις με πίνακες. Για παράδειγμα να “διαβαστούν” MxN αριθμοί σε έναν ορθογώνιο πίνακα “κατά στήλες” ή “κατά γραμμές”, να υπολογιστούν τα αθροίσματα των αριθμών κατά γραμμή ή κατά στήλη, να προσδιοριστεί το ελάχιστο κάθε γραμμής κλπ. Μια σημαντική δυσκολία έχει σχέση με τη φύση του πίνακα. Προκειμένου για πίνακες δυο διαστάσεων μπορεί να θεωρηθεί ότι μια ορθογώνια περιοχή μνημών ονομάζεται A κι άρα A(3,4) παραπέμπει σε μια συγκεκριμένη μνήμη από το σύνολο αυτών που ονομάσαμε A.



Οι διαστάσεις των πινάκων άνω των 2 θα καθοριστούν επακριβώς. Ο αριθμός των διαστάσεων αντιστοιχεί στον αριθμό των πληροφοριών που μας είναι απαραίτητες για να εντοπίσουμε με ακρίβεια ένα στοιχείο. Για παράδειγμα μέσα σε ένα βιβλίο, στη σελίδα 27, στη σειρά 4, το 6^ο γράμμα είναι ένα “ρ”. Επομένως χρειάζονται 3 πληροφορίες (3 αριθμοί) για να εντοπίσουμε το γράμμα αυτό. Αν λοιπόν τα γράμματα αυτού του βιβλίου καταχωρηθούν σε έναν πίνακα, ο πίνακας αυτός θα πρέπει να είναι τρισδιάστατος.

15. Δίνονται N τριάδες αριθμών. Κάθε τριάδα αντιστοιχεί σε ένα μαθητή και ο πρώτος αριθμός αντιστοιχεί στον κωδικό του, ο δεύτερος στο φύλο του (ο αν είναι άντρας και 1 αν είναι γυναίκα) και ο τρίτος είναι 1,2,3 ή 4, ανάλογα με τη δέσμη που διάλεξε. Να βρεθεί πόσα κορίτσια πήγαν σε κάθε δέσμη.

Η άσκηση παρουσιάζει ενδιαφέρον γιατί οι αρχάριοι προγραμματιστές χρησιμοποιούν κατά κανόνα ενδιάμεσες μεταβλητές για να ενημερώσουν τους μετρητές των δεσμών. Ένα τυπικό δείγμα είναι το εξής:

```
Αν DESMH=1 τότε METRHTHS_DESMHS=METRHTHS_DESMHS + 1
```

αντί για

```
METRHTHS_DESMHS(DESMH)=METRHTHS_DESMHS(DESMH) + 1
```

Οι εισαγωγικές αυτές ασκήσεις αντιστοιχούν σε μια σειρά εξακριβωμένων δυσκολιών που συναντούν οι αρχάριοι προγραμματιστές - κι οι οποίες μοιάζουν να είναι σχετικά ανεξάρτητες από τη χρησιμοποιούμενη γλώσσα προγραμματισμού και τη μέθοδο διδασκαλίας. Φυσικά οι ασκήσεις αυτές δεν είναι μοναδικές κι ούτε καλύπτουν όλο το φάσμα των δυσκολιών. Είναι γνωστό ότι παρόμοια προβλήματα παρουσιάζονται στην επεξεργασία των συμβολοσειρών (strings), στη διαχείριση δεικτών (pointers), στο πέρασμα παραμέτρων στις διαδικασίες και συναρτήσεις όπως και στην αναδρομή.

Η αντιμετώπιση όμως όλων αυτών των περιπτώσεων ξεπερνάει σαφώς τα όρια της παρούσας, που αποτελεί μια εισαγωγή στη διδακτική της Πληροφορικής.

10. ΠΑΡΑΡΤΗΜΑ 2: Η ΜΗΧΑΝΗ ΤΟΥ POST

Στα παρακάτω περιγράφεται ένα “μικροσκοπικό” περιβάλλον, βασισμένο στην μηχανή του E. Post, για μια εισαγωγή στις τυπικές (formal) μεθόδους προγραμματισμού και μάλιστα στο πλαίσιο ενός εισαγωγικού μαθήματος στον προγραμματισμό. Δίνονται παραδείγματα επίλυσης προβλημάτων στο περιβάλλον αυτό.

Εισαγωγή

Ο “συστηματικός προγραμματισμός” [Wirth 1975] απετέλεσε, όπως είναι γνωστό, μια πρόταση για αλλαγή του προγραμματιστικού στυλ, του μοντέλου δηλαδή της προγραμματιστικής δραστηριότητας στα τέλη της δεκαετίας του 1960. Η πρόταση αυτή, ουσιαστικά αυτή η *αλλαγή παραδείγματος* [10], υποστηρίχθηκε από πολλούς επιφανείς πληροφορικούς ([Dijkstra 1972], [Gries 1981], [Hoare 1969] μεταξύ των άλλων) - αν κι όχι απ’ όλους ([de Millo 1979], [Knuth 1973], [Knuth 1974]) - ως μια λύση στο αδιέξοδο στο οποίο είχε περιέλθει ο προγραμματισμός εξ αιτίας της λεγόμενης software crisis [Δαγδιλέλης 1996], εξ αιτίας δηλαδή του γεγονότος ότι το παραγόμενο λογισμικό είχε υπερβολικά μεγάλο κόστος και ταυτόχρονα υπερβολικά μικρή αξιοπιστία. Η επικράτηση του συστηματικού προγραμματισμού οφείλεται όχι στο γεγονός ότι συνηγόρησαν υπέρ αυτού ορισμένοι επιφανείς επιστήμονες, αλλά στο ότι ο συστηματικός προγραμματισμός απετέλεσε μια ουσιαστική λύση στο αδιέξοδο.

Έκτοτε ο συστηματικός προγραμματισμός επεκράτησε διεθνώς ως ένα σύνολο τυποποιημένων τεχνικών και μεθόδων για τον προγραμματισμό, δηλαδή ένα σύνολο τεχνικών και μεθόδων οι οποίες “συστηματοποιούν” την προγραμματιστική δραστηριότητα σε όλες τις συνιστώσες της: οι προδιαγραφές των προγραμμάτων, η κατασκευή και περιγραφή των αλγόριθμων κι η επικύρωση τους αντιμετωπίζονται πλέον με μια τυποποιημένη - κι αν είναι δυνατόν τυπική - μεθοδολογία. Ο τυπικός προγραμματισμός και γενικότερα οι φορμαλιστικές μέθοδοι θεωρήθηκαν ως το ανώτερο επίπεδο του συστηματικού προγραμματισμού δεδομένου ότι ο προγραμματισμός χάρη στις τυπικές μεθόδους θα μπορούσε να αναβιβασθεί σε μια επιστήμη όπως τα Μαθηματικά. Οι αντιρρήσεις υπήρξαν πολλές και ποικίλες [de Millo 1979]:

- α) ακόμη και τα Μαθηματικά είναι εκ πρώτης όψεως μια “επιστήμη του τυπικού” αλλά στην πραγματικότητα η κοινωνική τους διάσταση είναι πολύ σημαντική
- β) στα Μαθηματικά οι τυπικές μέθοδοι οδηγούν σε νέες γνώσεις (π.χ. θεωρήματα) ενώ στον προγραμματισμό θα καταλήγουν στην πιστοποίηση ορθής λειτουργίας ενός λογισμικού.
- γ) το μέγεθος και η πολυπλοκότητα των τυπικών αποδείξεων ακόμη και για μικρά

προγράμματα, καθιστούν πρακτικά ανεφάρμοστες τις τυπικές μεθόδους. Εξάλλου, για τον ίδιο λόγο, οι τυπικές αποδείξεις δεν αποτελούν εχέγγυα ορθής λειτουργίας (γιατί ακριβώς λόγω μεγέθους μπορεί κι οι ίδιες οι αποδείξεις να έχουν λάθη).

Σήμερα οι τυπικές μέθοδοι - αλλά με μια μη-αυστηρή τυπικότητα ανάλογη με εκείνη των σχολικών Μαθηματικών - διδάσκονται διεθνώς σε προπτυχιακό επίπεδο, αλλά χωρίς να εφαρμόζονται συστηματικά στην πράξη. Αυτό που πραγματοποιείται είναι η συστηματική παραγωγή προγραμμάτων καθώς και αποδείξεις με “χαλαρή” τυπικότητα - σκαριφήματα αποδείξεων μάλλον παρά ολοκληρωμένες αποδείξεις (για παράδειγμα με τρόπο συστηματικό προσδιορίζεται η αρχική και τελική κατάσταση του συστήματος σε κάθε διαδικασία και υποπρόγραμμα).

Όσοι διδάσκουν προγραμματισμό, στο σημείο αυτό βρίσκονται αντιμέτωποι με ένα ιδιαίτερο πρόβλημα: πώς μπορεί, στο πλαίσιο μιας διδασκαλίας, να δοθεί στις τεχνικές αυτές το πραγματικό τους νόημα (να καταστούν δηλαδή έμπρακτα φανεροί οι λόγοι για τους οποίους ο συστηματικός προγραμματισμός είναι αναγκαίος ([Δαγδιλέλης 1996]); Ένα δεύτερο, σχετικό με το πρώτο, ερώτημα είναι εκείνο του κόστους: οι τυπικές μέθοδοι προγραμματισμού έχουν πολύ υψηλό διανοητικό “κόστος” γιατί συχνά είναι μακρές κι επίπονες.

Μια κλασική λύση στο δεύτερο αυτό πρόβλημα αποτελεί η επινόηση “μικροσκοπικών” περιβαλλόντων - *μικροκόσμων* όπως ονομάζονται - οι οποίοι - λόγω μεγέθους των διαφόρων χαρακτηριστικών του περιβάλλοντος, όπως η “μικρή” γλώσσα προγραμματισμού και η φύση των προτεινομένων προβλημάτων - καθιστούν μικρότερες τις προτεινόμενες λύσεις.

Στα πλαίσια αυτά η μηχανή του Post αποτελεί ένα παράδειγμα τέτοιου περιβάλλοντος για εισαγωγικά μαθήματα στον τυπικό προγραμματισμό. Ο μικρόκοσμος τον οποίο περιγράφουμε παρουσιάζει το πλεονέκτημα της εξαιρετικής απλότητας. Όπως παρατηρεί και ο Usrensky [1983] η προσομοίωση του μπορεί να γίνει και με μια λωρίδα χαρτιού κι ένα συνδετήρα που θα παίζει το ρόλο του δρομέα. Εξάλλου ο μικρός αριθμός εντολών και η απλότητα τους καθιστούν τον μικρόκοσμό εξαιρετικά “οικονομικό” - η λειτουργία του μπορεί να διδαχθεί σε ελάχιστο χρόνο. Παρά ταύτα στο μικροσκοπικό αυτό περιβάλλον μπορούν να εφαρμοστούν πλήρως οι τυπικές μέθοδοι χωρίς να απαιτείται υπέρογκο διανοητικό κόστος για την κατανόηση και εφαρμογή τους. Για παράδειγμα ενώ εκ πρώτης όψεως πρόκειται για μια επιτακτική γλώσσα, είναι πολύ εύκολο να διαπιστώσει κανείς ότι οι εντολές λειτουργούν ως “μετασχηματιστές συμβολοσειρών” . Ο χρήστης λοιπόν (ο σπουδαστής) μπορεί πολύ σύντομα να “απομακρυνθεί “ από τη λογική της “μηχανής”, της κίνησης του δρομέα κλπ - αφαίρεση η οποία απαιτεί μεγάλη διανοητική προσπάθεια στις

συνήθεις επιτακτικές γλώσσες. Εξάλλου παρά την εξαιρετική του απλότητα, ο μικρόκοσμος με μερικές προθήκες μπορεί να επεκταθεί σε περισσότερα των δυο σύμβολα, σε περισσότερες διαστάσεις κλπ.

Σε ότι ακολουθεί περιγράφεται ένα “μικροσκοπικό” περιβάλλον, βασισμένο στην μηχανή του E. Post, για μια εισαγωγή στις τυπικές (formal) μεθόδους προγραμματισμού και μάλιστα στο πλαίσιο ενός εισαγωγικού μαθήματος στον προγραμματισμό. Στην 2η και 3η ενότητα περιγράφεται αναλυτικά το περιβάλλον. Στην 4η ενότητα δίνονται παραδείγματα επίλυσης προβλημάτων στο περιβάλλον αυτό. Οι διδακτικές χρήσεις της μηχανής του Post περιγράφονται στην 5η ενότητα ενώ η τελευταία ενότητα αποτελεί τον επίλογο.

Η μηχανή του Post

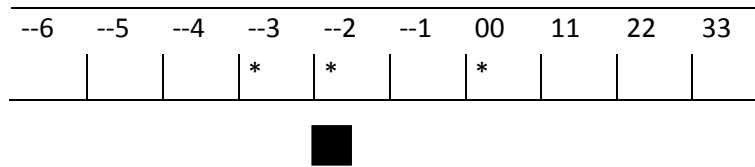
Στα 1936 ο Αμερικανός μαθηματικός Emil Post δημοσίευσε το άρθρο “Finite Combinatory process - formulation 1” [Post 1936]. Σ’ αυτό το άρθρο όπως και στο άρθρο “On computational numbers with an application of the Entscheidungsproblem” του A. Turing [1937] δίνονται οι πρώτοι ορισμοί της έννοιας του αλγόριθμου αλλά και διατυπώνονται θεωρητικά οι ιδιότητες των Η/Υ πριν ακόμη εμφανισθούν οι πρώτοι Η.Υ. Η μηχανή του Turing είναι δημοφιλέστερη της μηχανής του Post - παρά το γεγονός ότι σε πολλές περιπτώσεις η μηχανή του Post είναι απλούστερη καθόσον οι εντολές είναι απλούστερες και η τεχνική εγγραφής λιγότερο πολύπλοκη.

Η μηχανή του Post είναι μια ταινία αποτελούμενη από άπειρα διαδοχικά τετράγωνα που θα τα αποκαλούμε *κυψέλες (cells)*. Για την γραφική της απεικόνιση την σχεδιάζουμε οριζόντια και θεωρούμε ότι το πλήθος των κυψελών είναι απεριόριστο και κατά τις δυο κατευθύνσεις. Κάθε κυψέλη της ταινίας μπορεί να είναι κενή ή μπορεί να περιέχει ένα σύμβολο οπότε καλείται *κενή (empty)* ή *σημειωμένη (marked)* αντίστοιχα. Οι κυψέλες της ταινίας είναι αριθμημένες με ακεραίους αριθμούς ... -3, -2, -1, 0, 1, 2, 3, ...

Μια κεφαλή ανάγνωσης και εγγραφής που θα καλείται στο εξής *δρομέας (carriage)* κινείται αριστερά και δεξιά απέναντι από τις κυψέλες. Σε κάθε χρονική στιγμή που στο εξής θα καλείται *βήμα (step)*, ο δρομέας μπορεί να κινηθεί μόνο κατά μια κυψέλη αριστερά ή δεξιά. Ο δρομέας μπορεί να αλλάζει τη κατάσταση μιας κυψέλης σε κάθε βήμα, δηλαδή να την καθιστά σημειωμένη (αν είναι κενή), οπότε θεωρούμε ότι *γράφει στην κυψέλη (mark)* ή να την καθιστά κενή (αν είναι σημειωμένη), οπότε θεωρούμε ότι *διαγράφει (erase)*. Ακόμη ο δρομέας μπορεί να διαπιστώνει αν η εξεταζόμενη κυψέλη είναι κενή ή σημειωμένη και θεωρούμε ότι ο δρομέας *εξετάζει (examine)* τη συγκεκριμένη κυψέλη.

Ονομάζεται *εσωτερική κατάσταση της μηχανής (internal state)* του Post η πληροφορία που

αναφέρεται στα εξής : τις κυψέλες που είναι κενές, τις κυψέλες που είναι σημειωμένες και σε ποιά κυψέλη βρίσκεται σ' εκείνο το βήμα ο δρομέας. Στο Σχ. 1 δίνεται η γραφική απεικόνιση της μηχανής του Post σε μια ορισμένη εσωτερική κατάσταση, όπου έχουμε σημειωμένες δυο ομάδες κυψελών (τις κυψέλες -3, -2 και 0) και ο δρομέας βρίσκεται στην αριστερότερη κυψέλη της πρώτης ομάδας σημειωμένων κυψελών.



Σχήμα 1

Η λειτουργία της μηχανής του Post συνίσταται στην μετακίνηση του δρομέα κατά μήκος της ταινίας, στην δυνατότητα του δρομέα να γράφει ή να διαγράφει και να εξετάζει την κατάσταση μιας κυψέλης. Η λειτουργία αυτή πραγματοποιείται με την βοήθεια ενός προγράμματος που απαρτίζεται από ένα πεπερασμένο σύνολο εντολών. Οι εντολές αυτές είναι οι εξής :

- (α). Μετακίνηση του δρομέα μια κυψέλη δεξιά. Στο εξής θα παριστάνουμε την εντολή αυτή με το σύμβολο \rightarrow .
- (β). Μετακίνηση του δρομέα μια κυψέλη αριστερά. Στο εξής θα παριστάνουμε την εντολή αυτή με το σύμβολο \leftarrow .
- (γ). Σημείωση της θεωρούμενης κυψέλης (αν είναι κενή). Στο εξής θα παριστάνουμε την εντολή αυτή με το σύμβολο +.
- (δ). Διαγραφή της θεωρούμενης κυψέλης (αν είναι σημειωμένη). Στο εξής θα παριστάνουμε την εντολή αυτή με το σύμβολο -.
- (ε). Εξέταση της θεωρούμενης κυψέλης για το αν είναι κενή ή όχι. Μια κενή κυψέλη συμβολίζεται με b και μια σημειωμένη με *.
- (στ). Τερματισμός

Ένα **πρόγραμμα** [Post 1939, Uspensky 1983] για τη μηχανή του Post συνίσταται σε ένα πεπερασμένο σύνολο αριθμημένων εντολών 1, 2, 3, ..., n. Η εντολή με αριθμό i μπορεί να αντιστοιχεί σε μια από τις παρακάτω μορφές:

- i) Εκτέλεση της εντολής E_i (όπου $E_i = (\alpha), (\beta), (\gamma)$ ή (δ)) και στη συνέχεια εκτέλεση της εντολής με αριθμό j_i (goto j_i).
- ii) Εκτέλεση της εντολής (ε) και ανάλογα αν η απάντηση είναι Ναι ή Όχι εκτέλεση της εντολής j_i' ή j_i'' , όπου $j_i', j_i'' \in \{1, 2, 3, \dots, n\}$ (if b then goto j_i' else goto j_i'').

iii) Τερματισμός (εντολή (στ)).

Η εντολή τερματισμού του προγράμματος προφανώς δεν αλλάζει την εσωτερική κατάσταση της μηχανής αλλά συνεπάγεται *τερματισμό με εμφάνιση αποτελέσματος* (halt with result). Αν κατά την διάρκεια εκτέλεσης του προγράμματος από την μηχανή ζητηθεί να εκτελεσθεί εντολή με την οποία επιχειρείται να σημειωθεί μια κυψέλη που είναι ήδη σημειωμένη ή να διαγραφεί κυψέλη που είναι ήδη κενή τότε σταματά η περαιτέρω εκτέλεση του προγράμματος και θεωρούμε ότι συμβαίνει *λάθος χωρίς αποτέλεσμα* (error no result halt). Αν τέλος δοθεί εντολή που επαναλαμβάνει συνεχώς ένα σύνολο εντολών χωρίς τον τερματισμό τους τότε θεωρούμε ότι συμβαίνει *λάθος συνεχούς επανάληψης* (error infinite loop).

Για καθαρά διδακτικούς σκοπούς τροποποιούμε εν μέρει την μηχανή του Post. Οι τροποποιήσεις αυτές έχουν ως εξής:

1) Δεν θεωρούμε σύστημα συντεταγμένων για την ταινία. Η αναφορά στις κυψέλες γίνεται με διαφορετικό τρόπο που παρουσιάζεται στην επόμενη ενότητα.

2) Οι αρχικές εντολές της μηχανής του E. Post [ibid], όπως και η θεώρηση της μηχανής από τον Uspensky [14], δεν περιλαμβάνουν εντολές του τύπου while, repeat κλπ. Βασιζόμενοι στο γνωστό γεγονός ότι η εντολή GOTO δεν είναι απαραίτητη [Boehm 1966] δημιουργούμε, κατά προφανή τρόπο, εντολές του τύπου while κλπ. Για παράδειγμα η διαδικασία

While * do →

αντιστοιχεί στις εντολές

i if * then goto i+1 else goto i+2

i+1 → goto i

i+2

Από τυπική άποψη η while έχει την εξής σημασία:

αν ισχύει: A και B

κι εκτελεστεί η επαναληπτική διαδικασία { while B do something }

τότε, αν η επαναληπτική διαδικασία τερματιστεί, ισχύει: A και not B

3) Σε κάθε εντολή της μηχανής του Post, το δεύτερο μέρος της δίνει τον αριθμό της επόμενης εντολής που πρόκειται να εκτελεστεί (**goto xx**). Για λόγους απλότητας, θεωρούμε ότι η διαδοχή των εντολών καθορίζεται από τη θέση τους στο κείμενο κι έτσι οι εντολές εκτελούνται κατά την σειρά με την οποία είναι γραμμένες στο κείμενο - εκτός βέβαια από τις περιπτώσεις των επαναληπτικών διαδικασιών. Επιπλέον, το *σώμα* (body) των σύνθετων εντολών (για παράδειγμα το σώμα των επαναληπτικών εντολών) καθορίζεται από μια μετατόπιση του κειμένου.

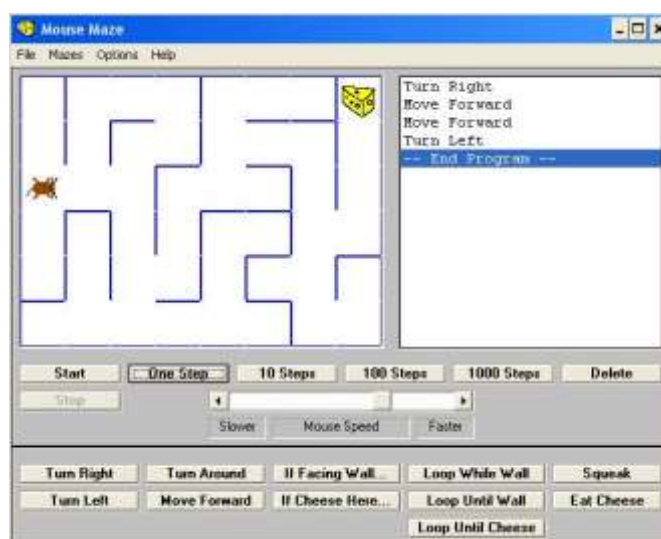
11. ΠΑΡΑΡΤΗΜΑ 3: MORTRAN

Το όνομα του συστήματος είναι πιθανόν ένας συνδυασμός από τις λέξεις Mouse και FORTRAN ή συνδυασμός των Mouse Relocation Translator.

Στο Διαδίκτυο μπορεί κανείς να βρει αναφορές στο Mortran 2 και Mortran 3. Πρόκειται για αναφορές σε ένα είδος συστήματος για μια λιγότερο "αυστηρή" έκδοση της FORTRAN (MORe forTRAN: δες για παράδειγμα:

http://rcwww.kek.jp/research/egs/docs/pdf/Elementary_Mortran3.pdf).

Η απλοποιημένη αυτή αναπαράσταση, επιτρέπει τη δόμηση ενός περιβάλλοντος, στο οποίο οι εντολές είναι προκαθορισμένες και αντιστοιχούν σε πλήκτρα (buttons), αποκλείοντας έτσι την πιθανότητα (τη δυνατότητα) συντακτικού λάθους. Ο χρήστης μπορεί να προσδιορίσει τη μορφή του λαβυρίνθου. Αξιοσημείωτο είναι επίσης το γεγονός, ότι η χρησιμοποιούμενη γλώσσα προγραμματισμού είναι προσανατολισμένη προς τη συγκεκριμένη εφαρμογή και όχι γενική: για παράδειγμα, η εντολή Loop Until Cheese είναι σαφώς προσανατολισμένη προς το περιβάλλον μέσα στο οποίο (υποτίθεται ότι) λειτουργεί το ποντίκι και όχι μια γενική επαναληπτική δομή Loop Until <condition>.



Εικόνα Α2.7: Το περιβάλλον MORTRAN

Το πρόβλημα της ανεύρεσης του δρόμου από το σημείο στο οποίο βρίσκεται το ποντίκι ως το τυρί, ανάγεται στην ανεύρεση ενός δρόμου (path) μέσα σε διδιάστατο λαβύρινθο – ένα πρόβλημα γνωστό στη θεωρία γραφημάτων. Ενδιαφέρουσες παραλλαγές περιλαμβάνουν την ύπαρξη «νησίδων» ή τον «αποκλεισμό» (την αδυναμία πρόσβασης) του στόχου. Αν και το πρόβλημα στη γενική του μορφή δεν παρουσιάζει βέβαια επιστημονικό ενδιαφέρον (η ανεύρεση δρόμου σε λαβύρινθο έχει ως λύση ένα γνωστό αλγόριθμο), ωστόσο το

περιβάλλον παρουσιάζει μερικά σημαντικά διδακτικά χαρακτηριστικά (για αρχαρίους προγραμματιστές), καθώς η γλώσσα προγραμματισμού αλλά και η διαχείρισή του περιβάλλοντος εργασίας είναι τετριμμένες – επομένως οι μαθητές μπορούν να τα μάθουν σε ελάχιστο χρονικό διάστημα – αλλά, ταυτόχρονα, οι μαθητές μπορούν να οδηγηθούν σταδιακά από τη μερική λύση στη γενική (δηλαδή στην ιδέα του αλγορίθμου).

Για παράδειγμα, το πρόβλημα της εύρεσης ενός αλγορίθμου που επιλύει όλους τους λαβυρίνθους – εκτός νησίδων και αποκλειομένων περιοχών – είναι ένα πολύ ενδιαφέρον θέμα (από διδακτική άποψη), το οποίο αποτελεί τη γενίκευση όλων των επιμέρους λύσεων για τους λαβυρίνθους, που μπορεί να επινοήσει ο χρήστης και φυσικά απαιτεί μια μακρόχρονη επεξεργασία από την πλευρά των αρχαρίων προγραμματιστών. Η αναζήτηση λοιπόν μιας γενικής λύσης, μπορεί να οδηγήσει σταδιακά στην κατανόηση της εννοίας του αλγορίθμου, ως γενικής μεθόδου επίλυσης μιας κλάσης προβλημάτων, έναντι των ad hoc λύσεων.

Συνοψίζοντας, το περιβάλλον Mortran έχει μερικά αξιοσημείωτα χαρακτηριστικά:

1. είναι ένα κλειστό σύστημα και η γλώσσα προγραμματισμού που ενσωματώνεται σε αυτό είναι εξαιρετικά περιορισμένη
2. περιλαμβάνει μια πολύ καλή διεπαφή που διευκολύνει το χρήστη και τον «απαλλάσσει» από το ενδεχόμενο των συντακτικών λαθών.
3. στηρίζεται σε μια μεταφορά (ποντίκι-λαβύρινθος-τυρί) που ξεφεύγει από το στερεότυπο των προβλημάτων που χρησιμοποιούνται σε εισαγωγικά μαθήματα προγραμματισμού.
4. παρά τη μινιμαλιστική του σχεδίαση, τα προβλήματα που μπορούν να επιλυθούν στο πλαίσιο του Mortran δεν είναι πάντα τετριμμένα. Μπορούν να κλιμακώνονται από πολύ απλά σε πολύ σύνθετα
5. σημαντικό (από διδακτική άποψη) είναι ότι οι λαβύρινθοι του Mortran μπορούν να κατηγοριοποιηθούν (τουλάχιστον ορισμένοι από αυτούς) και έτσι να αντιμετωπιστούν με ένα ενιαίο τρόπο. Με άλλα λόγια, οι λαβύρινθοι αυτοί μπορούν να κατηγοριοποιηθούν.
6. τα προβλήματα μπορούν να λυθούν είτε με «βηματική εξερεύνηση» (trial-and-error), είτε με τη χρήση αλγορίθμων που έχουν γενικότερη ισχύ. Τα προγράμματα που δημιουργούνται με trial-and-error μπορούν να αποδειχθούν πολύ «δαπανηρά» στην πράξη, όταν οι λαβύρινθοι είναι αρκετά πολύπλοκοι. Επιπλέον οι αλγόριθμοι Trial-and-error πρέπει να κατασκευάζονται από την αρχή, ακόμη και για όμοια προβλήματα.

Το περιβάλλον MORTRAN θα μπορούσε να χρησιμοποιηθεί για μια εισαγωγή στη

δημιουργία «γενικών» αλγορίθμων για την επίλυση κλάσεων προβλημάτων, αντί για την επίλυση προβλημάτων με τη μέθοδο trial-and-error.

Το πρόβλημα της δημιουργίας γενικών αλγορίθμων, μιας γενικής δηλαδή μεθόδου επίλυσης μιας κλάσης προβλημάτων, είναι ένα γνωστό πρόβλημα στη διδασκαλία του προγραμματισμού. Οι μαθητές είναι εξοικειωμένοι με το να εφαρμόζουν γνωστές τεχνικές και μεθόδους στην επίλυση προβλημάτων (αλγόριθμος επίλυσης μιας εξίσωσης, τύποι της Φυσικής) αλλά όχι με τη δημιουργία και την έκφραση μεθόδων που επιλύουν προβλήματα. Σε κάποιο βαθμό, το διδακτικό πρόβλημα που τίθεται, είναι ανάλογο με εκείνο που υφίσταται κατά τη μετάβαση από την αριθμητική στην Άλγεβρα, κατά την οποία οι μαθητές δυσκολεύονται να κατανοήσουν τη γενικότητα που προσφέρουν οι αλγεβρικές λύσεις έναντι των αριθμητικών μεθόδων.

Προτάσεις για τη δημιουργία διδακτικών σεναρίων με το MORTRAN

Φάση εξοικείωσης

Υποτίθεται ότι οι μαθητές σας έχουν λίγες γνώσεις προγραμματισμού.

Προτείνετε στους μαθητές μερικές στοιχειώδεις δραστηριότητες με σκοπό την εξοικείωση τους με τις λειτουργίες του περιβάλλοντος.

Επίλυση απλών προβλημάτων

Προτείνετε στους μαθητές μερικά προβλήματα προς επίλυση. Μερικά από αυτά θα πρέπει να είναι εξαιρετικά απλά και 1 ή 2 να είναι με πολύπλοκους λαβυρίθμους.

Οι μαθητές σας πιθανότατα, στην πλειοψηφία τους, θα επιλύσουν τα σχετικά προβλήματα με «βηματική κατασκευή» trial-and-error.

Συζητήστε με τους μαθητές σας το γεγονός ότι κάθε trial-and-error κατασκευή είναι πολύ χρονοβόρα και μοναδική και δείξτε τους τα ενδεχόμενα κοινά χαρακτηριστικά των απλών περιπτώσεων.

Αλγοριθμική αντιμετώπιση απλών προβλημάτων

Αναλύστε τη σημασία των προτεινομένων εντολών (Loop while wall κλπ) δίνοντας ενδεχομένως παραδείγματα χρήσης. Για παράδειγμα ο βρόχος

Loop while facing wall

Turn Left

End Loop

μας εξασφαλίζει ότι στην «έξοδο» του, ανεξάρτητα από την αρχικό προσανατολισμό του ποντικού, ο ποντικός θα είναι στραμμένος προς μια «ανοιχτή δίοδο» (χωρίς εμπόδιο ή εξωτερικό τοίχο).

Προτείνετε στους μαθητές σας να επιλύσουν τα προβλήματα MAZ1 και MAZ2 και προσανατολίστε τους προς τη δημιουργία ενός αλγορίθμου που αντιμετωπίζει και τους δυο λαβυρίνθους. Ζητήστε τους να επεκτείνουν η λύση τους και σε λαβυρίνθους όπως ο MAZ6 και δοκιμάστε τη λύση τους σε διάφορους λαβυρίνθους «του ιδίου τύπου».

«Κατάκτηση» προβλήματος σε απλούστερα προβλήματα I

Ζητήστε από τους μαθητές σας να αντιμετωπίσουν λαβυρίνθους και προβλήματα όπως τα easy 1,2,3.

Η ομοιότητα ανάμεσα στα προβλήματα είναι εμφανής – πρόκειται για τους ίδιους λαβυρίνθους με διαφορετικό σημείο άφιξης.

Δοκιμάστε να συζητήσετε με τους μαθητές σας τα κοινά χαρακτηριστικά των αλγορίθμων που επιλύουν τα προβλήματα αυτά: πρόκειται για «δυο γειτονικούς διαδρόμους». Στον ένα βρίσκεται το σημείο εκκίνησης (ο ποντικός) και στο άλλο το σημείο άφιξης (το τυρί)

Ένας αλγόριθμος και για τα τρία προβλήματα θα μπορούσε να περιγραφεί ως εξής (ως εντολές προς το ποντίκι):

Βρες το «άνοιγμα»

«Πέρασε» στο διπλανό διάδρομο

Πήγαινε στην άκρη του διαδρόμου

Στρίψε προς την αντίθετη κατεύθυνση

Βρες το τυρί.

Προσπαθήστε να βοηθήσετε τους μαθητές σας ώστε να επιλύσουν κάθε ένα από τα προβλήματα αυτά και να συνθέσουν την τελική λύση.

«Κατάκτηση» προβλήματος σε απλούστερα προβλήματα II

Με την ίδια τεχνική, της «κατάκτησης» ενός προβλήματος σε απλούστερα, προτείνετε στους μαθητές σας άλλες κατηγορίες λαβυρίνθων.

Ζητήστε τους να σκεφθούν και να επαληθεύσουν μήπως ορισμένοι αλγόριθμοι μπορούν να εφαρμοστούν και σε άλλες κατηγορίες λαβυρίνθων.

Συνθετότερα προβλήματα: 1

Ανοίξτε την εφαρμογή Mortran. Επιλέξτε το λαβύρινθο easy1. Επιλύστε το σχετικό πρόβλημα.

Το πρόγραμμα σας επιλύει και το πρόβλημα του λαβυρίνθου easy 2 καθώς και του λαβυρίνθου easy 3; Στην πραγματικότητα οι τρεις λαβύρινθοι έχουν ως κοινό χαρακτηριστικό ότι το σημείο άφιξης (τυρί) βρίσκεται στο «διπλανό» διάδρομο από αυτόν όπου βρίσκεται το ποντίκι. Οι τρεις λαβύρινθοι μπορούν να αντιμετωπιστούν με το ίδιο πρόγραμμα (η «λογική» του προγράμματος είναι: πέρασμα στο διπλανό διάδρομο, ξεκίνημα από έναν τοίχο (χωρίς να εξετάζεται το ενδεχόμενο να υπάρχει τυρί) και «σάρωση» του διαδρόμου μέχρι τον εντοπισμό του τυριού).

Συνθετότερα προβλήματα: 2

Ανοίξτε το Mortran. Επιλέξτε το λαβύρινθο Mark's Maze (στο μενού Mazes->all the rest)

Προσπαθήστε να επιλύσετε το πρόβλημα. Το πρόγραμμα σας επιλύει το πρόβλημα και των υπολοίπων λαβυρίνθων του μενού Mazes->all the rest; Αν το ποντίκι «ακολουθήσει» έναν τοίχο πάντοτε από την ίδια πλευρά (π.χ. τη δεξιά) και το τυρί βρίσκεται κοντά σε τοίχο, τότε το ποντικάκι αργά ή γρήγορα θα εντοπίσει το τυρί. Το πρόγραμμα δεν λειτουργεί σωστά στο λαβύρινθο Justin's Maze, γιατί εξ αιτίας του λαβυρίνθου και της θέσης του ποντικού, δεν έχει ισχύ ο αλγόριθμος της «ίδιας πλευράς» που αναφέρεται παραπάνω.

Ενότητα ΘΕΩΡΙΑ ΔΙΔΑΚΤΙΚΗΣ ΤΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ 2

Διδακτικά σενάρια – τι είναι και πώς δημιουργούνται

ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΕΠΙΜΟΡΦΩΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Το σενάριο περιλαμβάνει 2 μέρη:

Στο πρώτο μέρος μελετάται το θεωρητικό πλαίσιο ΓΤΠΠ.

Στο δεύτερο μέρος, σε συνδυασμό με το πλαίσιο του πρώτου μέρους, παρουσιάζεται ένα μοντέλο για τη δημιουργία διδακτικών σεναρίων

ΕΚΤΙΜΩΜΕΝΗ ΔΙΑΡΚΕΙΑ ΕΠΙΜΟΡΦΩΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

3 διδακτικές ώρες. Είναι ενδεχόμενο να απαιτηθεί συμπληρωματική εργασία περίπου μιας ώρας.

ΣΚΟΠΟΙ ΚΑΙ ΣΤΟΧΟΙ ΤΟΥ ΕΠΙΜΟΡΦΩΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Ο γενικός σκοπός είναι η γνώση του θεωρητικού πλαισίου ΓΤΠΠ και του [προτεινόμενου μοντέλου για τα διδακτικά σενάρια](#).

Ειδικότερα αναμένεται οι επιμορφούμενοι με το πέρας της Συνεδρίας (και της πρόσθετης εργασίας αν απαιτηθεί):

- (1) Να έχουν κατανοήσει τις βασικές αρχές και τη χρησιμότητα των διδακτικών σεναρίων
- (2) Να έχουν εξοικειωθεί με την ορολογία και τα επιμέρους στοιχεία του διδακτικού σεναρίου
- (3) Να αποκτήσουν την ικανότητα να εντοπίζουν τα στοιχεία αυτά σε ήδη δημιουργημένα σενάρια και να έχουν μια πρώτη εξάσκηση στη δημιουργία διδακτικών σεναρίων για ένα (σχετικά) σύντομο σενάριο της επιλογής τους.

ΘΕΩΡΗΤΙΚΕΣ ΕΝΝΟΙΕΣ ΑΠΟ ΤΗ ΔΙΔΑΚΤΙΚΗ ΤΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

Το μοντέλο της Γνώσης Τεχνολογικής Παιδαγωγικής Περιεχομένου (ΓΤΠΠ – TRACK)

Η έννοια της ΓΤΠΠ

Η έννοια της **Γνώσης Τεχνολογικής Παιδαγωγικής Περιεχομένου** (ΤΠΠΠ – ενίοτε στα ελληνικά αναφέρεται και ως Τεχνολογική, Παιδαγωγική Γνώση Περιεχομένου) διατυπώθηκε από τους Mishra και Koehler (2006) με στόχο να περιγράψουν ολοκληρωμένα το πλαίσιο των παραγόντων που καθορίζουν την ένταξη των ΤΠΕ στη σχολική τάξη. Το μοντέλο αυτό δεν αντιμετωπίζει ανεξάρτητα το Περιεχόμενο, την Παιδαγωγική και τα Τεχνολογικά μέσα αλλά μέσα από το σύνθετο σύστημα αλληλο-συσχετίσεων που ορίζουν οι τρεις αυτές παράμετροι.

Από την εισαγωγή της ως θεωρητική έννοια, η ΤΠΠΠ έχει μετασηματιστεί σε ένα χρήσιμο πλαίσιο-εργαλείο για την υποστήριξη της ένταξης των ΤΠΕ στη σχολική πρακτική προσφέροντας πολλές δυνατότητες στην εκπαιδευτική έρευνα, στην εκπαίδευση και επαγγελματική ανάπτυξη των εκπαιδευτικών και στην ενσωμάτωση των ΤΠΕ στην τάξη.

Οι Mishra και Koehler (2006) πρότειναν την έννοια της ΤΠΠΠ, η οποία δεν αντιμετωπίζει ανεξάρτητα τις τρεις συστατικές συνιστώσες (Τεχνολογία, Περιεχόμενο, Παιδαγωγική) αλλά εστιάζει στις μεταξύ τους αλληλεπιδράσεις, στο πλαίσιο που διαμορφώνουν τα διάφορα μαθησιακά περιβάλλοντα των ΤΠΕ.

Έτσι ορίζονται τρεις νέες μορφές γνώσης:

Παιδαγωγική Γνώση: Η ΠΓ περιλαμβάνει το σώμα της γνώσης που αφορά στην αναπαράσταση, περιγραφή και μετασηματισμό των επιστημονικών εννοιών και διαδικασιών, τις παρανοήσεις, προϋπάρχουσες γνώσεις ή γνωστικές δυσκολίες των μαθητών, σε παιδαγωγικές στρατηγικές και τεχνικές που είναι αποτελεσματικές στην πράξη, κ.λπ.

Γνώση Περιεχομένου: Η ΤΠΠ στη γλώσσα, τα μαθηματικά και τις φυσικές επιστήμες περιλαμβάνει ζητήματα που αφορούν στο πώς οι επιστημονικές έννοιες και διαδικασίες μετασηματίζονται σε συγκεκριμένα τεχνολογικά περιβάλλοντα. Για παράδειγμα, οι αλλαγές στη φύση της επιστήμης που φέρνουν οι ΤΠΕ, νέες μέθοδοι και εργαλεία που χρησιμοποιούνται για να λύσουν τα προβλήματα των Φυσικών Επιστημών, οι μέθοδοι μοντελοποίησης της επιστημονικής γνώσης, η χρήση λογισμικών προσομοίωσης (π.χ. στη φυσική ή στη χημεία) κ.λπ. Τα λογισμικά δυναμικής γεωμετρίας και τα λογισμικά σχεδίασης

διαφοροποιούν τους τρόπους με τους οποίους οικοδομείται η μαθηματική γνώση, οι επεξεργαστές κειμένου και οι κειμενικές βάσεις δεδομένων αλλάζουν τον τρόπο με τον οποίο προσεγγίζουμε τη διδασκαλία της γλώσσας.

Τεχνολογική Γνώση: Η ΤΓ στα βασικά γνωστικά αντικείμενα περιλαμβάνει τη γνώση για το πώς οι ΤΠΕ μπορούν να υποστηρίξουν συγκεκριμένες παιδαγωγικές στρατηγικές στην τάξη, π.χ. να ενθαρρύνουν τη διερευνητική ή τη συνεργατική μάθηση.

Οι Koehler και Mishra (2006) υποστηρίζουν ότι η ουσιαστική ένταξη των ΤΠΕ στην πράξη απαιτεί την κατανόηση από τους εκπαιδευτικούς των σχέσεων μεταξύ των τριών συνιστωσών της ΤΠΓΠ. Παρότι το πλαίσιο αυτό εμφανίζεται ως ένα απλό στη σύλληψή του και λογικό θεωρητικό κατασκεύασμα, είναι δύσκολο να κατανοηθεί από τους εκπαιδευτικούς και ακόμη δυσκολότερο να εφαρμοστεί αποτελεσματικά στην εκπαιδευτική πρακτική).

Συνιστώσες της ΤΠΓΠ	Άξονες γνώσεων & ικανοτήτων
Παιδαγωγική Γνώση	Επιστημονική γνώση Πρόγραμμα Σπουδών των επιμέρους γνωστικών αντικειμένων Μετασχηματισμός της επιστημονικής γνώσης Μαθησιακές δυσκολίες και παρανοήσεις των μαθητών (σε συγκεκριμένες ενότητες ή έννοιες) Μαθησιακές στρατηγικές Παιδαγωγικές και διδακτικές στρατηγικές Εκπαιδευτικό πλαίσιο
Γνώση Περιεχομένου	Τεχνολογικά μέσα και εργαλεία διαθέσιμα για συγκεκριμένα γνωστικά αντικείμενα Δεξιότητες χειρισμού και τεχνικές δεξιότητες σχετικά με συγκεκριμένες έννοιες και γνώσεις των φυσικών επιστημών, της γλώσσας και των μαθηματικών Μετασχηματισμός της επιστημονικής γνώσης με ΤΠΕ Επιστημονική μέθοδος και ΤΠΕ
Τεχνολογική Γνώση	Μαθησιακές και διδακτικές στρατηγικές βασισμένες σε ΤΠΕ

	Πρώθηση επιστημονικής διερεύνησης με ΤΠΕ Υποστήριξη καλλιέργειας τεχνολογικών δεξιοτήτων Μαθησιακή υποστήριξη με ΤΠΕ (scaffolding) Χειρισμός τεχνικών δυσκολιών σε υπολογιστικά λογισμικά και περιβάλλοντα
--	--

Έτσι, άμεση διδασκαλία και η εκμάθηση λογισμικών ή εργαλείων, κατά τη διάρκεια των μαθημάτων στο ΚΣΕ προτείνεται να είναι σχετικά περιορισμένη. Η εξοικείωση με τα τεχνολογικά εργαλεία και η πρόσκτηση των σχετικών δεξιοτήτων θα επιτευχθεί με την εμπλοκή των επιμορφούμενων στην επίλυση νοηματοδοτούμενων διδακτικών προβλημάτων μέσω του σχεδιασμού αυθεντικών σεναρίων και μαθησιακών δραστηριοτήτων βασισμένων σε ΤΠΕ, οι οποίες να εδράζονται σε ένα σαφές παιδαγωγικό υπόβαθρο.

Αυτό διότι σύμφωνα με τη θεωρία αυτή, είναι ο συνδυασμός των τριών ειδών γνώσεων που μπορεί να είναι διδακτικά αποδοτικός – όχι η κάθε μία συνιστώσα χωριστά..

Το εκπαιδευτικό σενάριο με ΤΠΕ

Η έννοια του εκπαιδευτικού σεναρίου

Τα **εκπαιδευτικά σεναρία** που θα χρησιμοποιηθούν στην επιμόρφωση είναι στενά συνδεδεμένο με το εκπαιδευτικό υλικό που συνοδεύει ένα υπολογιστικό περιβάλλον και πολλές φορές έχουν αναπτυχθεί από την ομάδα ανάπτυξης του εκπαιδευτικού λογισμικού. Επιπρόσθετα, πολλά υπολογιστικά περιβάλλοντα που μπορούν να χρησιμοποιηθούν για εκπαιδευτικούς σκοπούς (όπως επεξεργαστές κειμένου, λογισμικά διαδικτύου, κλπ.) δεν διαθέτουν έτοιμα σεναρία, γεγονός που καθιστά αναγκαία την ικανότητα του εκπαιδευτικού να δημιουργεί κατάλληλες εκπαιδευτικές δραστηριότητες. Σε αυτή την περίπτωση, η ύπαρξη εκπαιδευτικών σεναρίων ή τουλάχιστον διδακτικών δραστηριοτήτων είναι απαραίτητη. Δεδομένης της συνθετότητας της εκπαιδευτικής πράξης αλλά και των ιδιαίτερων αναγκών της κάθε τάξης, είναι συχνά αναγκαίο να μπορεί η εκπαιδευτικός να προσαρμόζει ή και να δημιουργεί τα δικά της εκπαιδευτικά σεναρία. Στο πλαίσιο αυτό, η ανάπτυξη της ικανότητας εκ μέρους της εκπαιδευτικού να προσαρμόζει υπάρχοντα ή να δημιουργεί νέα εκπαιδευτικά σεναρία και δραστηριότητες με ΤΠΕ ώστε να καλύπτει τις πραγματικές διδακτικές της ανάγκες συνιστά κομβικό σημείο της επιμόρφωσης.

Ένα εκπαιδευτικό σενάριο μπορεί να είναι τμήμα του υποστηρικτικού υλικού για τον/την εκπαιδευτικό (στο οποίο συνήθως περιγράφεται η ιδέα του σεναρίου, ο σκοπός, οι στόχοι

του, κλπ) και του έντυπου ή άλλου υλικού για το μαθητή (στο οποίο συμπεριλαμβάνεται η διδακτική δραστηριότητα που έχει συνήθως τη μορφή «φύλλου δραστηριότητας» ή εργασίας). Όλο και πιο συχνά όμως, κυρίως για ανοικτού τύπου υπολογιστικά περιβάλλοντα, είναι δυνατόν να δημιουργηθούν από τους ίδιους τους εκπαιδευτικούς νέα εκπαιδευτικά σενάρια και διδακτικές δραστηριότητες. Στο πλαίσιο της επιμόρφωσης, μάλιστα, η ικανότητα αυτή συνιστά βασική δραστηριότητα για τους εκπαιδευτικούς, οι οποίοι πρέπει να καταστούν ικανοί στο πώς να δημιουργούν νέα σενάρια και διδακτικές δραστηριότητες με διάφορες κατηγορίες λογισμικού. Είναι αυτονόητο ότι η ανάπτυξη της ικανότητας αυτής είναι συνυφασμένη - και συχνά αναπτύσσεται - με την προσαρμογή υπαρχόντων σεναρίων και διδακτικών δραστηριοτήτων, τα οποία είναι πλέον διαθέσιμα στις ψηφιακές βιβλιοθήκες, στις δικές τους εκπαιδευτικές ανάγκες και στις ανάγκες των μαθητών τους.

Ένα **Εκπαιδευτικό ή Διδακτικό Σενάριο** με Τεχνολογίες της Πληροφορίας και των Επικοινωνιών (ΤΠΕ) περιγράφει το σύνολο των διδακτικών δραστηριοτήτων και των χρησιμοποιούμενων εργαλείων (συμβολικών, όπως σχήματα ή λογισμικά και φυσικών, όπως ειδικές κατασκευές) που συνιστούν το σημείο εκκίνησης καθώς και το γενικότερο πλαίσιο μέσα στο οποίο λαμβάνουν χώρα δραστηριότητες διδασκαλίας και μάθησης. Στο πλαίσιο της επιμόρφωσης ένα εκπαιδευτικό σενάριο κάνει χρήση των ΤΠΕ και ειδικότερα εκπαιδευτικών περιβαλλόντων μάθησης με υπολογιστές. Το εκπαιδευτικό σενάριο συνεπώς αφορά εκπαιδευτικούς και μαθητές, κάνει χρήση κατάλληλων διδακτικών στρατηγικών και αποσκοπεί στην επίτευξη ενός μαθησιακού αποτελέσματος μέσω της χρήσης κατάλληλου υπολογιστικού περιβάλλοντος (εκπαιδευτικό λογισμικό ή και υλικό). Συνήθως, το σενάριο αποσκοπεί στη διδασκαλία και τη μάθηση μιας ή περισσότερων βασικών εννοιών ενός γνωστικού αντικείμενου μέσα από το υφιστάμενο πρόγραμμα σπουδών. Επιπρόσθετα, το σενάριο μπορεί να προσεγγίζει διαθεματικά ή διεπιστημονικά έννοιες από διάφορα γνωστικά αντικείμενα, ενώ μπορεί επίσης να αφορά μάθηση εννοιών εκτός αναλυτικού προγράμματος.

Το εκπαιδευτικό σενάριο περιέχει οδηγίες για τους εκπαιδευτικούς, το θεωρητικό πλαίσιο μέσα στο οποίο εντάσσεται η προβληματική του, τα απαιτούμενα υλικά υλοποίησής του, «φύλλα» δραστηριοτήτων για τους μαθητές και ενδεχομένως άλλο υλικό (κατασκευές, έντυπο υλικό, αρχεία λογισμικών, κλπ.). Ένα εκπαιδευτικό σενάριο υλοποιείται συνεπώς από μια σειρά διδακτικών δραστηριοτήτων. Το σενάριο, με άλλα λόγια, είναι μια πλήρης διδακτική παρέμβαση με σκοπό, στόχους, προβληματική, διαδικασία εφαρμογής μέσω κατάλληλων δραστηριοτήτων και διδακτικών στρατηγικών, διαδικασία αξιολόγησης, κλπ.

Οι βασικές προδιαγραφές ποιότητας ενός εκπαιδευτικού σεναρίου σύμφωνα με τις σύγχρονες θεωρήσεις για τη γνώση και τη μάθηση ακολουθούν πέντε μεγάλους άξονες:

- α) την προβληματική του σεναρίου,
- β) το περιεχόμενο και τη μορφή του σεναρίου,
- γ) την ακολουθούμενη διδακτική μεθοδολογία,
- δ) τις ακολουθούμενες διδακτικές στρατηγικές και
- ε) την αξιοποίηση των ΤΠΕ στη μαθησιακή διαδικασία.

Α) Σε ότι αφορά στην **προβληματική των εκπαιδευτικών σεναρίων** σύμφωνα με τις σύγχρονες παιδαγωγικές θεωρίες (όπως είναι οι εποικοδομιστικές και οι κοινωνικοπολιτισμικές προσεγγίσεις), θα πρέπει κατά περίπτωση:

- να αναδεικνύουν το δυνατό εύρος και τη μεγάλη ποικιλία των μέσων και των υπηρεσιών που προσφέρουν οι ΤΠΕ στην Εκπαίδευση και ειδικότερα την προστιθέμενη αξία που έχει η ψηφιακή τεχνολογία στην εκπαιδευτική διαδικασία,
- να αναδεικνύουν τα κριτήρια επιλογής ποιοτικά κατάλληλων εκπαιδευτικών λογισμικών ή τουλάχιστον αποφυγής αναποτελεσματικών εκπαιδευτικών λογισμικών εστιάζοντας στην προστιθέμενη αξία των πρώτων,
- να προωθούν τη διεπιστημονική προσέγγιση εννοιών και μεθόδων με την υποστήριξη που παρέχουν οι ΤΠΕ
- να αναδεικνύουν τη σπουδαιότητα και τη δυναμική των κοινοτήτων μάθησης που μπορούν να δημιουργηθούν με την αξιοποίηση των δυνατοτήτων του Διαδικτύου και ιδιαίτερα των εφαρμογών και των υπηρεσιών Web 2.0 στο πλαίσιο της θεμελίωσης του σχολείου της Κοινωνίας της Γνώσης.

Β) Σε ότι αφορά στο περιεχόμενο και τη μορφή των εκπαιδευτικών σεναρίων, θα πρέπει κατά περίπτωση:

- να εξειδικεύουν σε αντικείμενα (π.χ. φυσικές επιστήμες, μαθηματικά, γλώσσα, κλπ.) και επιμέρους τμήματα του Αναλυτικού Προγράμματος όπου οι ΤΠΕ μπορούν να παίξουν ρόλο γνωστικού εργαλείου,
- να υποστηρίζουν διερευνητικού και ανακαλυπτικού τύπου μαθησιακές καταστάσεις (σε όλα τα γνωστικά αντικείμενα),
- να ευνοούν δραστηριότητες επίλυσης προβλήματος, λήψης απόφασης και ανάπτυξης της κριτικής σκέψης (σε όλα τα γνωστικά αντικείμενα),

- να υποστηρίζουν δραστηριότητες συμβολικής έκφρασης, επικοινωνίας και αναζήτησης πληροφοριών (σε όποια αντικείμενα αυτό είναι απαραίτητο, όπως γλώσσα, γραπτή έκφραση, κλπ.).

Γ) Σε ότι αφορά στην ακολουθούμενη **διδακτική μεθοδολογία**, τα εκπαιδευτικά σενάρια θα πρέπει να προωθούν, να ενισχύουν και να ενθαρρύνουν:

- την ανάπτυξη κατάλληλων γνώσεων και δεξιοτήτων των εκπαιδευτικών ώστε να είναι σε θέση να διαχειριστούν επιτυχώς εκπαιδευτικές εφαρμογές (δεδομένων των αλλαγών που επιφέρει η χρήση των ΤΠΕ στην τριμελή σχέση μαθητή – εκπαιδευτικού - γνώσης)
- τη διαμόρφωση κριτικής στάσης των εκπαιδευτικών απέναντι στη διδακτική χρήση των ΤΠΕ (αφού όλα τα υπολογιστικά περιβάλλοντα δεν είναι ισοδύναμα από διδακτική και μαθησιακή άποψη)
- τον προβληματισμό για το μετασχηματισμό του ρόλου των κύριων συντελεστών μιας μαθησιακής διαδικασίας (μαθητής, εκπαιδευτικός, γνώση, υπολογιστικό περιβάλλον) και τις μεταβολές στο έργο των εκπαιδευτικών, όπως για παράδειγμα το πώς διαμεσολαβείται η γνώση ή το πώς πρέπει να αξιολογείται το μαθησιακό αποτέλεσμα όταν χρησιμοποιούνται οι ΤΠΕ.

Δ) Σε ότι αφορά στις ακολουθούμενες **διδακτικές στρατηγικές**, τα εκπαιδευτικά σενάρια όχι απλώς πρέπει να διευκολύνουν τη χρήση των ΤΠΕ στη διδακτική και τη μαθησιακή διαδικασία, αλλά πρέπει επίσης να ευνοούν και να προωθούν νέες, εναλλακτικές μορφές διδασκαλίας που είναι περισσότερο συμβατές με τις σύγχρονες παιδαγωγικές και διδακτικές θεωρίες και με τη χρήση των ΤΠΕ. Πιο συγκεκριμένα πρέπει να υποστηρίζουν και να προωθούν τη μετάβαση:

- από τη μετωπική διδασκαλία, στη διδασκαλία με ομάδες και τη συνεργατική μάθηση,
- από τη δασκαλοκεντρική διδασκαλία όπου οι ΤΠΕ χρησιμοποιούνται ως εποπτικό μέσο, στη μαθητοκεντρική διδασκαλία κατά την οποία οι ΤΠΕ χρησιμοποιούνται ως γνωστικό εργαλείο,
- από τη διάλεξη ως διδακτική μέθοδο στη διερευνητική και την ανακαλυπτική μέθοδο,
- από την παθητική, σε μια κινητοποιημένη σχολική τάξη μέσα από την ενεργητική συμμετοχή, την επικοινωνία μεταξύ μαθητών και τις αυθεντικές δραστηριότητες,

- από μεθόδους αξιολόγησης του μαθητή που βασίζονται στο αποτέλεσμα μιας και μόνης τελικής δοκιμασίας σε μεθόδους που βασίζονται σε διαδικασίες και παραγόμενα προϊόντα και όχι μόνο,
- από ένα σύστημα μάθησης στο οποίο όλοι μαθαίνουν τα ίδια πράγματα, σε ένα σύστημα όπου ενδεχομένως ο καθένας μαθαίνει διαφορετικά πράγματα,
- από τους γνωστούς τρόπους επικοινωνίας (κυρίως προφορικής ή γραπτής) σε τρόπους επικοινωνίας που ενσωματώνουν πολλαπλές αναπαραστάσεις, εικόνες, κείμενα, σύμβολα, χάρτες πολλαπλών, συνδεδεμένων και ταυτόχρονων αναπαραστάσεων κ.ά.

Ε) Σε ότι αφορά στη χρήση των ΤΠΕ, είναι προφανές ότι ένα εκπαιδευτικό σενάριο που αξιοποιεί τις ΤΠΕ μπορεί να ευνοήσει την ανάπτυξη ικανοτήτων (σύνολο γνώσεων και δεξιοτήτων) υψηλού επιπέδου από τους μαθητές, όπως

- Ικανότητα επίλυσης προβλημάτων,
- Ανάπτυξη της κριτικής σκέψης,
- Ικανότητα διερεύνησης και αναζήτησης πληροφοριών σε ένα ευρύ φάσμα δεδομένων,
- Ανάπτυξη δεξιοτήτων λήψης απόφασης,
- Δυνατότητα μοντελοποίησης φαινομένων και καταστάσεων των πραγματικού κόσμου,
- Ικανότητα συνεργασίας και από κοινού προσέγγισης και επίλυσης προβλημάτων,
- Διεπιστημονική προσέγγιση της γνώσης,
- Ανάπτυξη δεξιοτήτων μεταφοράς γνώσεων από ένα πλαίσιο σε ένα άλλο.

Η έμφαση, με άλλα λόγια, των εκπαιδευτικών σεναρίων δίνεται στην υλοποίηση διδακτικών καταστάσεων που ευνοούν την ανάπτυξη από τους μαθητές γνωστικών ικανοτήτων υψηλού επιπέδου, που κατά τεκμήριο είναι εγκάρσιες στο πρόγραμμα σπουδών, όπως επίλυση προβλήματος, πειραματική διαδικασία, δραστηριότητες διερεύνησης και ανακάλυψης, μοντελοποίηση, διεπιστημονική προσέγγιση, λήψη απόφασης, κριτική σκέψη, αναστοχασμός και νέος κριτικός γραμματισμός.

Τα τμήματα ενός διδακτικού σεναρίου στην Πληροφορική

Συνοπτικά, ένα «διδακτικό σενάριο» αποτελεί μια όσο το δυνατόν πιο πλήρης περιγραφή ενός μαθήματος που προορίζεται να χρησιμοποιηθεί και από άλλα άτομα – εκτός από το δημιουργό του. Η βασική λειτουργία του διδακτικού σεναρίου είναι διπλή:

(α) αποτελεί ένα σύνολο σημείων τα οποία πρέπει να λάβει υπόψη του ο εκπαιδευτικός όταν σχεδιάζει ένα νέο μάθημα. Το σημείο αυτό είναι σημαντικό. Ο εκπαιδευτικός, όταν σχεδιάζει ένα μάθημα, για παράδειγμα για τις βάσεις δεδομένων, θα πρέπει να λάβει υπόψη του πολλούς παράγοντες: τους λόγους για τους οποίους διδάσκεται το συγκεκριμένο μάθημα, το «ρόλο» που παίζει το μάθημα αυτό μέσα στο σύνολο των μαθημάτων Πληροφορικής που διδάσκεται στη συγκεκριμένη βαθμίδα και τύπο σχολείου (για παράδειγμα: επαγγελματικό Λύκειο), τις ενδεχόμενες δυσκολίες που αντιμετωπίζουν οι μαθητές όταν διδάσκονται το μάθημα αυτό. Προφανώς είναι πρακτικά πολύ δύσκολο ένας εκπαιδευτικός να έχει αυτές τις γνώσεις για το σύνολο των μαθημάτων που διδάσκει. Το σενάριο λοιπόν του δίνει την ευκαιρία να στοχαστεί πάνω στις παραμέτρους αυτές. Μπορεί να μη γνωρίζει όλες τις απαντήσεις, έχει όμως επίγνωση των ερωτημάτων που τίθενται. Εξάλλου, η αύξηση των διαφόρων δικτύων εκπαιδευτικών, σημαίνει ότι εύκολα μπορεί κανείς να εντοπίσει και να χρησιμοποιήσει ένα σενάριο άλλου συναδέλφου.

(β) αποτελεί μια περιγραφή ενός μαθήματος δομημένη έτσι ώστε να περιγράφει όσο το δυνατόν περισσότερες όψεις της διδασκαλίας, σε μορφή τέτοια ώστε να μπορεί να χρησιμοποιηθεί και από άλλους εκπαιδευτικούς.

Μερικές βασικές αρχές που σχετίζονται με τα διδακτικά σενάρια:

1. Όταν πρόκειται να παρουσιαστεί μια νέα έννοια πρέπει, στο μέτρο του δυνατού, να προτείνονται προβλήματα που καθιστούν την νέα έννοια ένα είδος εργαλείου που επιτρέπει την επίλυση του προβλήματος
2. Ένα σημείο εκκίνησης μπορεί να προέρχεται από την ίδια την ιστορία της έννοιας που πρόκειται να διδαχθεί
3. Καλό είναι να λαμβάνονται υπόψη οι (ενδεχόμενες) λανθασμένες αντιλήψεις των μαθητών και να προτείνονται προβλήματα που οδηγούν σε αδιέξοδο (γνωστική σύγκρουση). Μη διορθώνετε ένα λάθος, αν ο μαθητής δεν έχει καταλάβει ότι είναι λάθος. Βοηθείστε τον να αντιληφθεί το λάθος δίνοντας του εργαλεία για να το εντοπίσει (μεθόδους, τεχνικές,...) και αν ο μαθητής καταλήξει σε αδιέξοδο, τότε δώστε του βοήθεια.

4. Να λαμβάνετε υπόψη το συγκεκριμένο περιβάλλον εργασίας – προφανώς πολλά πράγματα αλλάζουν: για παράδειγμα στη Logo είναι εύκολο να διακρίνει κανείς το σωστό από το λάθος (όταν εργάζεται με σχήματα). Ορισμένα είδη προβλημάτων επιλύονται ευκολότερα σε κάποια περιβάλλοντα – για παράδειγμα η αναδρομή είναι πιο εύκολη στη Logo από τις «τυπικές» γλώσσες προγραμματισμού. Αντίθετα η διαχείριση της εισόδου και εξόδου δεδομένων είναι ευκολότερη στις συνηθισμένες εμπορικές και επαγγελματικές γλώσσες.
5. Η επιλογή ενός ή περισσότερων περιβαλλόντων είναι αποφασιστικής σημασίας: Σχεδιάστε μακροπρόθεσμα, ξεκαθαρίστε τους στόχους σας και ακολουθείτε μια σταθερή γραμμή «εργασίας» – π.χ. πάντοτε επαλήθευση των αλγορίθμων «με το χέρι», στο χάρτινο υπολογιστή
6. Πόσο «πλήρες» πρέπει να είναι ένα σενάριο; Δηλαδή πρέπει να περιλαμβάνει όλα τα «τμήματα» και τις προσεγγίσεις; Η απάντηση είναι «όχι». Πρέπει, όταν δημιουργείται ένα σενάριο, να τίθενται όλα τα στοιχεία που είναι γνωστά. Ωστόσο, κάθε φορά που προστίθεται ένα νέο στοιχείο, θα πρέπει το σενάριο να επικαιροποιείται.

Παρατίθενται τα βασικά τμήματα ενός σεναρίου. Η λεπτομερέστερη ανάλυσή τους πραγματοποιείται σε ενότητες που ακολουθούν.

1. Τίτλος

Θα πρέπει να είναι κατατοπιστικός κατά το δυνατόν (έστω και αν είναι λίγο μακροσκελής).

2. Εκτιμώμενη διάρκεια

Για παράδειγμα: δύο διδακτικές ώρες – προσοχή ! ο υπολογισμός των ωρών θα πρέπει να είναι (κατά το δυνατόν) ρεαλιστικός. Υπάρχει γενικά μια τάση υποτίμησης του απαιτούμενου χρόνου.

3. Ένταξη στο πρόγραμμα σπουδών

Σε ποια σχολική τάξη (ή σχολικές τάξεις) εντάσσεται; Σε ποια βαθμίδα και σε ποιο τύπο σχολείου. Σε ποιο πρόγραμμα σπουδών;

4. Σκοποί και στόχοι

(ενδεχομένως και μεταστόχοι). Οι στόχοι θα πρέπει να είναι ρεαλιστικοί, όχι βερμπαλιστικοί και επακριβώς προσδιορισμένοι. Για παράδειγμα, δε μπορεί να είναι στόχος μιας δώωρης διδασκαλίας το παρακάτω: «*οι μαθητές να μάθουν να προγραμματίζουν*». Επίσης, οι στόχοι

δε μπορεί να είναι πολυάριθμοι: μια διδασκαλία 3 ωρών δε μπορεί, για παράδειγμα, να έχει 8 στόχους – στην περίπτωση αυτή ή οι στόχοι είναι υπερβολικά γενικευμένοι και αόριστοι ή ανέφικτου.

5. Συνοπτική περιγραφή της διδασκαλίας

Περιγράφεται (με συνοπτικό ή επεκτεταμένο τρόπο) το σενάριο

6. Επιστημολογική προσέγγιση και εννοιολογική ανάλυση

Πολλές φορές (αν όχι κατά κανόνα) η δημιουργία νέων μεθόδων ή εννοιών και η εξέλιξη τους είναι συνέπεια των εξελίξεων στον αντίστοιχο επιστημονικό κλάδο ή των κοινωνικών πρακτικών που συνδέονται με αυτήν. Είναι για παράδειγμα γνωστό ότι ο δομημένος προγραμματισμός καθιερώθηκε μέσα από μια μακρόχρονη και έντονη διαμάχη στους κύκλους των επιστημόνων και επαγγελματιών της Πληροφορικής. Η ύπαρξη πολλών λαθών, οφειλομένων στον άναρχο τρόπο με τον οποίο εργαζόταν οι Πληροφορικοί μέχρι τις αρχές της δεκαετίας του '70, καθιστούσε αναξιόπιστο και πανάκριβο το λογισμικό που κατασκευαζόταν. Ο δομημένος προγραμματισμός υπήρξε λοιπόν ένα εργαλείο για την αντιμετώπιση του προβλήματος αυτού. Επίσης, πολλές αλγοριθμικές τεχνικές έχασαν την πρακτική τους αξία, με την εξέλιξη τη τεχνολογίας, ενώ παρουσιάστηκε ανάγκη για άλλες. Αυτές οι συνθήκες δε μπορούν βέβαια να παρουσιαστούν μέσα στην τάξη. Ο εκπαιδευτικός ωστόσο, αν τις γνωρίζει, μπορεί να οργανώσει το μάθημα του έτσι ώστε να καταστήσει φανερή την αξία των εννοιών αυτών. Ακόμη, ο εκπαιδευτικός, εφόσον το γνωρίζει, θα πρέπει να λαμβάνει υπόψη του το ενδεχόμενο ύπαρξης αναλόγων εννοιών σε άλλα μαθήματα ή ακόμη και στην καθημερινότητα. Η ύπαρξη τέτοιων διασυνδέσεων μπορεί να είναι πολύ παραγωγική (από διδακτική σκοπιά), αλλά μπορεί να είναι και ανασταλτική. Ένα παράδειγμα αυτού του είδους αποτελεί η έννοια της μεταβλητής: συνδέεται με την έννοια της μεταβλητής στα Μαθηματικά, αλλά και με την έννοια των παραμέτρων και με την έννοια της κυψέλης στα λογιστικά φύλλα. Οι διαφορές και ομοιότητες μεταξύ τους μπορεί να είναι ένα διδακτικό υποστήριγμα για τον εκπαιδευτικό.

Η εννοιολογική ανάλυση μπορεί να οδηγήσει σε μια διεύρυνση των εννοιών που διδάσκονται (ο ρόλος των διασυνδεδεμένων γνώσεων μπορεί να είναι παραγωγικός): για παράδειγμα οι μαθητές κατανοούν τη λειτουργία του δυαδικού συστήματος, αλλά μπορεί κανείς να προκαλέσει μια βαθύτερη κατανόηση αν αναζητήσει αναλογίες, ομοιότητες και διαφορές με το δεκαδικό σύστημα: με ποιο τρόπο αναγνωρίζεται ένα «ζυγός» (άρτιος) αριθμός στο δεκαδικό σύστημα; Στο δυαδικό; Πως εξηγείται αυτό; Όλοι οι αριθμοί του δεκαδικού συστήματος μπορούν να μετατραπούν στο δυαδικό (η ερώτηση είναι βέβαια

διατυπωμένη με λανθασμένο τρόπο γι να διευκολυνθεί η κατανόησή της από τους μαθητές); Μπορούν να υπάρξουν κλάσματα στο δυαδικό σύστημα; Αντίστοιχα βέβαια, η έκφραση $X=X-1$ (τουλάχιστον με το συμβολισμό αυτό) δημιουργούσε μια μικρή σύγχυση καθώς είναι μεν έγκυρη προγραμματιστικά, αλλά όχι αν εκληφθεί ως Μαθηματική ισότητα.

7. Επεκτάσεις/διασυνδέσεις των εννοιών ή των δραστηριοτήτων

Πρόκειται για την ανάλυση της "θέσης" της προς διδασκαλία έννοιας στο πλαίσιο του μαθήματος. Πρέπει να υπάρχει σύνδεση ανάμεσα στην έννοια και στα όσα ακολουθούν και στα οποία θα χρειαστεί η έννοια ή στα όσα έχουν προηγηθεί. Ο εκπαιδευτικός πρέπει να λαμβάνει υπόψη του την "θέση" του προς διδασκαλία αντικειμένου τόσο μέσα στην επιστήμη και τη σχολική της έκδοση όσο και την παρουσία της σε ένα τεχνολογικό περιβάλλον: π.χ. η έννοια της αναδρομής και των αναδρομικών διαδικασιών. Κάθε έννοια συνδέεται άμεσα με μερικές άλλες, τουλάχιστον στη σχολική πραγματικότητα. Καλό λοιπόν είναι να λαμβάνονται υπόψη και πιθανές επεκτάσεις της έννοιας. Πώς η συγκεκριμένη δραστηριότητα μπορεί να αποτελέσει απαρχή για άλλες επιμέρους δραστηριότητες που επεκτείνουν την αρχική; Μπορεί να χρησιμοποιηθεί η μέθοδος ή το αποτέλεσμα για κάποιο άλλο πρόβλημα ή δραστηριότητα; Μπορεί να ενεργοποιηθεί ένας προβληματισμός των μαθητών με αφορμή τη συγκεκριμένη δραστηριότητα προς μια περαιτέρω εμβάθυνση; Για παράδειγμα ένα εισαγωγικό πρόβλημα για τη διδασκαλία των δομών επανάληψης, πολύ συχνά μπορεί να επεκταθεί σε παρόμοια προβλήματα, ή προβλήματα με το ίδιο περιεχόμενο, αλλά διαφορετική διατύπωση.

8. Πολλαπλές αναπαραστάσεις – πολλαπλές προσεγγίσεις.

Πολύ συχνά οι έννοιες στην Πληροφορική – για παράδειγμα στον προγραμματισμό - έχουν πολλαπλά πλαίσια εκφοράς ή αναπαραστάσης: ένα πρόγραμμα μπορεί να μελετηθεί κατά την εκτέλεση του, από τον αλγόριθμο που είναι γραμμένος στο χαρτί ή σε ένα σύστημα προσομοίωσης. Ένα διδακτικό σενάριο πρέπει να λαμβάνει υπόψη του αυτή τη διάσταση. Κάθε ένα από τα πλαίσια αυτά έχει τα μειονεκτήματα και τα πλεονεκτήματά του: η προσομοίωση της εκτέλεσης ενός αλγορίθμου στο χαρτί, «με το χέρι», δίνει πολλές πληροφορίες στον προγραμματιστή, αλλά φυσικά είναι εξαιρετικά χρονοβόρα και δαπανηρή σε προσπάθεια. Σε ορισμένες περιπτώσεις μάλιστα, όταν το πρόγραμμα είναι μεγάλου μεγέθους, η προσομοίωση αυτή είναι πρακτικά αδύνατη. Ο διδάσκων θα πρέπει να εκτιμήσει, ανάλογα με την περίπτωση, ποια από τα πλαίσια αυτά θα χρησιμοποιεί κατά τη διδασκαλία του, προκειμένου να καταστήσει αποτελεσματικότερη τη διδασκαλία.

9. Πρόβλεψη δυσκολιών.

Το σενάριο θα πρέπει να ενσωματώνει τις «συνήθειες», συστηματικές δυσκολίες των μαθητών. Για παράδειγμα, οι περισσότεροι μαθητές κάνουν το γνωστό λάθος:

```
EMVADON:=VASIS * YPSOS;
```

```
readln (VASIS);
```

```
Readln (YPSOS);
```

```
Writeln EMVADON;
```

Αν $VASIS=8$ και $YPSOS=4$, τότε ποιο είναι το αποτέλεσμα της εκτέλεσης του;

Ο εκπαιδευτικός δεν πρέπει να αποφεύγει προβλήματα αυτού του είδους, αλλά να τα χρησιμοποιήσει ως ένα διδακτικό μέσο που θα βοηθήσει τους ίδιους τους μαθητές να κατανοήσουν το λάθος τους – και ενδεχομένως τα αίτια του.

Για παράδειγμα, στο παραπάνω πρόβλημα, ο καθηγητής δεν θα πρέπει να «παρουσιάσει» το πρόγραμμα και να εξηγήσει το λάθος, αλλά θα πρέπει να ζητήσει από τους μαθητές να προβλέψουν το αποτέλεσμα της εκτέλεσής του: τι θα συμβεί αν δώσω τιμές 8 και 4 στις μεταβλητές VASIS και YPSOS (πιθανή απάντηση των μαθητών: 32); Αν δώσω 4 και 8 (πάλι 32). Αν δώσω 10 και 0; Στη συνέχεια, όταν το πρόγραμμα εκτελεστεί, ο καθηγητής, στο μέτρο που του τα επιτρέπει ο διδακτικός χρόνος και η οργάνωση του μαθήματος, δεν πρέπει να εξηγήσει ο ίδιος ο λάθος, αλλά να τους προτρέψει με προσομοίωση στο χαρτί να εντοπίσουν οι ίδιοι το λάθος.

10. Γιατί να χρησιμοποιηθεί ο υπολογιστής

Το καινοτόμο περιβάλλον δεν αποτελεί από μόνο του λόγο για να υλοποιηθεί μια διδακτική δραστηριότητα με τη χρήση τεχνολογίας. Οι επιλογές θα πρέπει να αξιολογούνται όχι με βάση τον καινοτόμο χαρακτήρα τους, αλλά την εκτιμώμενη διδακτική τους αποτελεσματικότητα. Πρέπει λοιπόν η απόφαση αυτή να δικαιολογείται από το γεγονός ότι η χρήση της τεχνολογίας θα επιτρέψει την υλοποίηση επιθυμητών ενεργειών που δεν θα μπορούσαν να υλοποιηθούν στο παραδοσιακό περιβάλλον. Πρέπει να καθίσταται φανερή η συμβολή της τεχνολογίας στην επίτευξη συγκεκριμένων διδακτικών στόχων. Βέβαια η Πληροφορική, ως μάθημα, είναι απολύτως συνυφασμένη με την ψηφιακή τεχνολογία και τους Η.Υ. ωστόσο, η επιλογή των κατάλληλων λογισμικών και περιβαλλόντων επηρεάζει τη διδασκαλία. Το σενάριο θα πρέπει να αναδεικνύει την ορθολογική χρήση των ΤΠΕ μέσα από το προσδοκώμενο «διδακτικό κέρδος». Προσδοκάται η συμβολή στο γνωστικό επίπεδο ή ανάπτυξη μιας συγκεκριμένης δεξιότητας; Μήπως η χρήση λογισμικού μπορεί- ανάλογα με την περίπτωση - να δημιουργεί πρόσθετα προβλήματα; Απαιτείται πχ ένα χρονικό διάστημα

για την εξοικείωση των μαθητών με ένα νέο περιβάλλον εργασίας και ενδεχομένως απαιτούνται πόροι και υποδομή που δεν είναι διαθέσιμοι (όπως π.χ. ένας βιντεοπρωτζέκτορας ή ένας Η/Υ ανά μαθητή, πρόσβαση στο Διαδίκτυο στο σπίτι). Ταυτόχρονα, πιθανόν να επέλθει ένας περιορισμός στην εικόνα που έχουν οι μαθητές για τη συγκεκριμένη έννοια. Μήπως είναι πρόωρη η εισαγωγή της αν δεν έχει εξασφαλιστεί άλλη σχετική γνώση ή δεξιότητες; Υπάρχει κίνδυνος να δημιουργηθεί παρανόηση για κάποιο θέμα στους μαθητές; Είναι ενδεχόμενη μια απώλεια σε θέματα δεξιοτήτων;

Ταυτόχρονα, αξιολογώντας τις διδακτικές προθέσεις και τις δυνατότητες ενός λογισμικού το σενάριο πρέπει να λαμβάνει υπόψη του θέματα όπως: Χρησιμοποιούνται περισσότερα από ένα λογισμικά ή το ίδιο λογισμικό με πολλούς τρόπους; Χρησιμοποιείται το Διαδίκτυο; Ποιοι λόγοι υπαγορεύουν τη χρήση καθενός λογισμικού;

11. Διδακτικός θόρυβος.

Ένα ακόμη ζητούμενο είναι η μείωση του «*διδακτικού θορύβου*», όρος που αναφέρεται στις ανεπιθύμητες σε εξωτερικούς παράγοντες ή σε παράπλευρες δραστηριότητες (πχ οι υπερβολικά μακροσκελείς υπολογισμοί), που μπορούν εξ ολοκλήρου να επισκιάσουν τα πραγματικά αντικείμενα του μαθήματος. Η λειτουργία του εργαστηρίου για παράδειγμα, μπορεί να αποτελέσει ένα σημαντικό παράγοντα στη ροή και οργάνωση του μαθήματος. Μια βλάβη αποτελεί έναν αστάθμητο παράγοντα που δεν μπορεί να προβλεφθεί. Ο χρόνος που χρειάζεται το εργαστήριο για να είναι σε πλήρη ετοιμότητα και λειτουργία είναι γνωστός και επηρεάζει το διδακτικό χρόνο. Οι προηγούμενες γνώσεις των μαθητών, για παράδειγμα, μπορούν επίσης να επηρεάσουν τη ροή του μαθήματος. Ο εκπαιδευτικός δηλαδή, μπορεί να υποθέτει ότι οι μαθητές έχουν κάποιες γνώσεις από προηγούμενα μαθήματα (ή ακόμη και από άλλες τάξεις), αλλά αυτό να μην είναι ορθό – κυρίως αν πρόκειται για έννοιες ή μεθόδους που θεωρούνται «*δύσκολες*» από τους μαθητές. Για παράδειγμα, οι έννοιες τις οποίες αντιμετωπίζουμε στη διδακτική διαδικασία έχουν για τους μαθητές μια «*διπλή ζωή*». Αρχικά αποτελούν αντικείμενο μάθησης (π.χ. οι μαθητές διδάσκονται την έννοια της βρόχου, τι είναι, πώς επιλέγουμε την κατάλληλη δομή). Αργότερα, δεν αποτελεί πια το αντικείμενο μάθησης, αλλά το μέσο προκειμένου να ασχοληθούμε με άλλα αντικείμενα (πχ τον υπολογισμό ενός μέσου όρου). Ακόμη και ως μέσο όμως, η επιλογή του βρόχου μπορεί να δημιουργεί ανεπιθύμητο διδακτικό θόρυβο που πρέπει να ληφθεί υπόψη στη σχεδίαση του μαθήματος – δηλαδή ενώ ο εκπαιδευτικός σχεδιάζει ένα μάθημα για να διδάξει έναν αλγόριθμο ταξινόμησης (για παράδειγμα), διαπιστώνει ότι ορισμένοι μαθητές δεν έχουν

κατανοήσει πλήρως τη λειτουργία του βρόχου *όσο...επανάλαβε*. Η σχεδίαση ενός μαθήματος μπορεί να λαμβάνει υπόψη της παράγοντες αυτού του είδους.

12. Χρήση εξωτερικών πηγών.

Είναι σημαντική παράμετρος το να γνωρίζει ο επιμορφωτής-εκπαιδευτικός από πού μπορεί να αντλήσει πρόσθετες πληροφορίες για την προς διδασκαλία έννοια (π.χ. από το Διαδίκτυο), πού θα βρει - ενδεχομένως - πρόσθετο διδακτικό υλικό, σημειώσεις, αναφορές από παρόμοιες διδασκαλίες κλπ (π.χ. πού θα ψάξει στο Διαδίκτυο;)

13. Υποκείμενη θεωρία μάθησης (ή υποκείμενες θεωρίες).

Συνδέεται με τον τρόπο που θεμελιώνεται η οργάνωση του μαθήματος. Μπορεί, για παράδειγμα, μια δραστηριότητα εκγύμνασης (drill and practice), ενταγμένη στο πλαίσιο της επίλυσης προβλήματος, καθοδηγούμενη από το δάσκαλο να μετατραπεί πιο κοντά σε μια κατά προσέγγιση πολύ πιο «ανοιχτή». Πώς τεκμηριώνεται η θεμελίωση του μαθήματος σε κάποια θεωρία (ή σε κάποιες θεωρίες); Ο εκπαιδευτικός λοιπόν πρέπει να θεμελιώνει το σχεδιασμό σε κάποιες διδακτικές θεωρίες, να δίνει στο μαθητή τη δυνατότητα να εξερευνήσει, να ανταλλάξει απόψεις και να συνεργαστεί με τους συμμαθητές του. Ποιος είναι στην περίπτωση αυτή ο ρόλος του ίδιου του εκπαιδευτικού;

14. Επισήμανση μικρομεταβολών στην οργάνωση του μαθήματος και στο νόημα των εννοιών.

Το γεγονός ότι μια έννοια είναι διαμεσολαβημένη από τη διεπαφή δίνει στο χρήστη την αίσθηση ότι διαχειρίζεται άμεσα έναν μικρόκοσμο ενώ στην πραγματικότητα διαχειρίζεται κατά τρόπο έμμεσο μια συγκεκριμένη υλοποίηση προσομοίωσης ενός συστήματος. Οι μαθητές θεωρούν, για παράδειγμα, ότι οι αριθμοί που χειρίζονται στο Excel ή σε ένα προγραμματιστικό περιβάλλον είναι οι ακέραιοι ή οι πραγματικοί αριθμοί. Είναι δυνατόν να δίνεται έμφαση σε κάποιες όψεις της έννοιας ενώ αποκρύπτονται ή υποβαθμίζονται άλλες.

15. Διδακτικό συμβόλαιο

Είναι το σύνολο των συμπεριφορών του διδάσκοντος που αναμένονται από το μαθητή και το αντίστοιχο σύνολο των συμπεριφορών του μαθητή που αναμένονται από το δάσκαλο. Υπάρχει ενδεχόμενο σε κάποιο σημείο της ροής της δραστηριότητας να ανατραπεί το διδακτικό συμβόλαιο. Το συμβόλαιο αυτό δεν είναι ρητά εκφρασμένο και γίνεται αντιληπτό κάθε φορά που με κάποιο τρόπο ανατρέπεται. Αυτό πρέπει να επισημαίνεται. Για παράδειγμα οι αρχάριοι προγραμματιστές δίνουν πολύ μεγαλύτερη σημασία στον τρόπο

εμφάνισης των αποτελεσμάτων – παρά στα αποτελέσματα. Στην έννοια αυτή θα υπάρξει και άλλη, εκτενέστερη αναφορά σε ενότητες που θα ακολουθήσουν.

16. Οργάνωση τάξης – εφικτότητα σχεδίασης.

Ο ρόλος εκπαιδευτικού είναι καθοριστικός στο σημείο αυτό. Ανάλογα με τις διδακτικές του προθέσεις καλείται να κάνει μια διαχείριση του αριθμού των μαθητών, των διαθέσιμων υπολογιστών, του τρόπου εργασίας (ατομικά – ομαδικά), του χρόνου (πόσες διδακτικές ώρες και με τι επιμέρους στόχους κάθε φορά). Το μάθημα είναι πραγματοποιήσιμο ή ανέφικτο; (πχ η χρήση 4 διαφορετικών λογισμικών σε μια διδακτική ώρα είναι εξαιρετικά δύσκολη για πρακτικούς λόγους). Και κυρίως, λαμβάνει υπόψη του τις πραγματικές συνθήκες διεξαγωγής του μαθήματος;

17. Περιγραφή και ανάλυση των φύλλων εργασίας (ή και άλλου διδακτικού υλικού) – αξιολόγηση των μαθητών

Ενδεχόμενες κατηγορίες δραστηριοτήτων που μπορούν να περιέχουν τα φύλλα εργασίας είναι οι εξής::

α) Δραστηριότητες ψυχολογικής και γνωστικής προετοιμασίας. η διαμόρφωση κατάλληλου συναισθηματικού κλίματος στην τάξη, η διαμόρφωση κινήτρου για το μάθημα, η ενημέρωση για τον σκοπό και τους στόχους του μαθήματος καθώς και η αποτίμηση της υπάρχουσας γνώσης, η ανίχνευση των γνωστικών δυσκολιών και των αναπαραστάσεων των μαθητών.

β) Δραστηριότητες «διδασκαλίας» του γνωστικού αντικειμένου (για παράδειγμα πως περιγράφεται ή λειτουργεί κάποια συγκεκριμένη δομή) ή επίλυση κάποιου προβλήματος. Προτεινόμενες διδακτικές στρατηγικές και βοήθειες που περιέχει το σενάριο για τη διδασκαλία του γνωστικού αντικειμένου. Αιτιολογούνται οι λόγοι για τους οποίους ο εκπαιδευτικός επέλεξε τις εν λόγω διδακτικές στρατηγικές και πως εντάσσονται στο ή στα φύλλα εργασίας. Πρέπει να αναφέρεται που εμπλέκονται και με τρόπο εμπλέκονται τα χρησιμοποιούμενα υπολογιστικά περιβάλλοντα

γ) Δραστηριότητες εμπέδωσης του γνωστικού αντικειμένου (επαναληπτικές δραστηριότητες, ανάλογα προβλήματα, επεκτάσεις). Χρησιμοποιείται κάποιο λογισμικό ή ψηφιακό περιβάλλον (ενδεχομένως διαφορετικό από αυτό που χρησιμοποιήθηκε στις προηγούμενες φάσεις της διδασκαλίας);

δ) Δραστηριότητες αξιολόγησης του γνωστικού αντικειμένου (με σκοπό καθαρά την αξιολόγηση του αποτελέσματος της διδασκαλίας). Χρησιμοποιείται κάποιο λογισμικό ή ψηφιακό περιβάλλον (ενδεχομένως διαφορετικό από αυτό που χρησιμοποιήθηκε στις

προηγούμενες φάσεις της διδασκαλίας); Για την αξιολόγηση, ο εκπαιδευτικός θα πρέπει να αποφασίσει για το είδος της (κλειστού τύπου, με αντικειμενικές ερωτήσεις, ανοιχτού τύπου κλπ. Ο εκπαιδευτικός

ε) Μεταγνωστικές δραστηριότητες. Χρησιμοποιείται κάποιο λογισμικό ή ψηφιακό περιβάλλον (ενδεχομένως διαφορετικό από αυτό που χρησιμοποιήθηκε στις προηγούμενες φάσεις της διδασκαλίας);

Φάσεις ανάπτυξης εκπαιδευτικού σεναρίου

Όπως είδαμε στην προηγούμενη ενότητα, η ανάπτυξη εκπαιδευτικών σεναρίων με ΤΠΕ συνιστά μια πρωτότυπη και ιδιαίτερα σύνθετη παιδαγωγική δραστηριότητα, η οποία απαιτεί γνώσεις που αφορούν

α) το προς διδασκαλία γνωστικό αντικείμενο,

β) τη διδακτική του γνωστικού αντικειμένου,

γ) τις παιδαγωγικές και τις ψυχολογικές θεωρίες για τη διδασκαλία και τη μάθηση και

δ) τη χρησιμοποιούμενη τεχνολογία και ειδικότερα την προστιθέμενη αξία που μπορεί να προσδώσει η τεχνολογία αυτή στη διδακτική και τη μαθησιακή διαδικασία.

Στο πλαίσιο αυτό, τα διδακτικά σενάρια δομούνται γύρω από 2 άξονες:

Συνοπτική παρουσίαση του Σεναρίου (τίτλος, εκτιμώμενη διάρκεια, εμπλεκόμενες γνωστικές περιοχές, τάξεις, συμβατότητα με το πρόγραμμα σπουδών, οργάνωση της διδασκαλίας, γνωστικά προαπαιτούμενα & απαιτούμενη υλικοτεχνική υποδομή, διδακτικοί Στόχοι (ως προς το γνωστικό αντικείμενο, ως προς τη χρήση των ΤΠΕ, ως προς τη μαθησιακή διαδικασία).

Διδακτική προσέγγιση (θεωρητική και μεθοδολογική προσέγγιση, διδακτική προσέγγιση με τις ΤΠΕ, η προβληματική του σεναρίου, διδακτικές δραστηριότητες, αξιολόγηση και πιθανές επεκτάσεις).

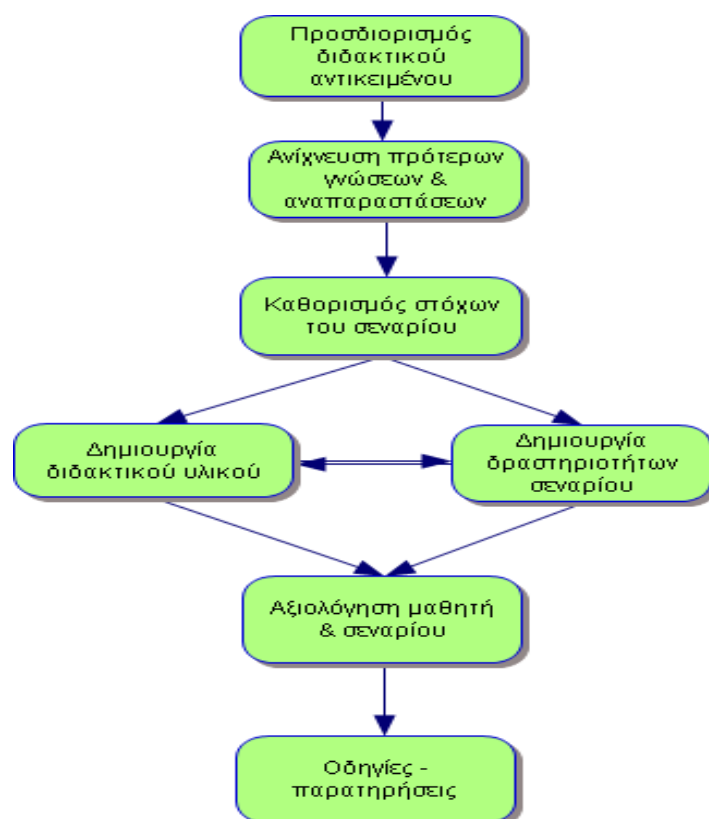
Στην ενότητα αυτή προτείνεται ένα ενδεικτικό μοντέλο σχεδίασης εκπαιδευτικών σεναρίων με ΤΠΕ που μπορεί να εφαρμοστεί στην πρωτοβάθμια και τη δευτεροβάθμια εκπαίδευση, το οποίο αναπλαισιώνει την παραπάνω δομή ανάπτυξης σεναρίων, τεκμηριώνοντας με μεγαλύτερη λεπτομέρεια αφενός το σκεπτικό πάνω στο οποίο πρέπει να βασίζεται ένα σενάριο που κάνει χρήση των ΤΠΕ (Φάσεις Α έως Δ και Στ, βλέπε παρακάτω) και αφετέρου

τον τρόπο με τον οποίο σχεδιάζονται οι δραστηριότητες στην τάξη (Φάση Ε, βλέπε παρακάτω).

Το μοντέλο αυτό δεν πρέπει να θεωρηθεί ότι συνιστά μια αποκλειστική μέθοδο που πρέπει να ακολουθείται κατά γράμμα από την εκπαιδευτικό που επιδιώκει να σχεδιάσει μαθησιακές δραστηριότητες με ψηφιακές τεχνολογίες. Αποτελεί επί της ουσίας ένα πλαίσιο μέσα στο οποίο θα μπορούσε να κινηθεί η εκπαιδευτικός ώστε να οργανώσει τις διδακτικές της παρεμβάσεις με γνώμονα τα ευρήματα της διδακτικής των επιστημών και την κατάλληλη χρήση της τεχνολογίας, δίνοντας έμφαση στην προστιθέμενη αξία της.

Είναι προφανές ότι, τουλάχιστον στο πλαίσιο της εποικοδομιστικής και της κοινωνικοπολιτισμικής προσέγγισης για τη διδασκαλία και τη μάθηση, δεν υπάρχει ένας συγκεκριμένος αλγόριθμος δημιουργίας εκπαιδευτικών σεναρίων. Η ποικιλομορφία της εκπαιδευτικής πρακτικής και η συνθετότητα των διαδικασιών διδασκαλίας και μάθησης δεν επιτρέπουν έναν πλήρη και αναλυτικό διδακτικό σχεδιασμό ή προγραμματισμό (δηλαδή έναν αλγόριθμο με τη μορφή βημάτων, όπως συνήθως προτείνει η συμπεριφοριστική προσέγγιση), τον οποίο η εκπαιδευτικός ακολουθεί κατά γράμμα όταν πραγματοποιεί μια διδακτική παρέμβαση. Εντούτοις, τα εκπαιδευτικά σενάρια, σε γενικές γραμμές εμφανίζουν μία σταθερή δομή, και συνακόλουθα τόσο η σχεδίαση και η ανάπτυξή τους όσο και η πραγματοποίησή τους σε συνθήκες τάξης μπορεί να ακολουθήσει μια σειρά από διακριτές φάσεις. Οι φάσεις αυτές αποτυπώνουν διαφορετικά στιγμιότυπα της προετοιμασίας ή της δημιουργίας των δραστηριοτήτων του σεναρίου.

Στο προτεινόμενο μοντέλο, η ανάπτυξη ενός εκπαιδευτικού σεναρίου με ΤΠΕ συμπεριλαμβάνει τουλάχιστον τις ακόλουθες – σε στενή συνήθως μεταξύ τους σχέση και αλληλεπίδραση – επτά (7) **φάσεις** (σχήμα Α1.2).



Σχήμα 1: φάσεις ανάπτυξης εκπαιδευτικού σεναρίου

- A. Το **διδακτικό αντικείμενο** του εκπαιδευτικού σεναρίου (τίτλος, τάξη (π.χ. Α΄ Γυμνασίου, Β΄ Λυκείου), εμπλεκόμενες γνωστικές περιοχές, γνωστικά προαπαιτούμενα, κλπ.)
- B. Οι **αναπαραστάσεις** των μαθητών και οι πιθανές δυσκολίες της σκέψης τους σχετικά με το γνωστικό αντικείμενο
- Γ. Οι **διδακτικοί στόχοι** του εκπαιδευτικού σεναρίου (ως προς το γνωστικό αντικείμενο, ως προς τη χρήση των ΤΠΕ, ως προς τη μαθησιακή διαδικασία)
- Δ. Το **διδακτικό υλικό** του εκπαιδευτικού σεναρίου και η απαιτούμενη υλικοτεχνική υποδομή
- Ε. Η οργάνωση της διδασκαλίας στη βάση κατάλληλων **δραστηριοτήτων υλοποίησης** του εκπαιδευτικού σεναρίου στην τάξη (διδακτικές προσεγγίσεις και στρατηγικές, αξιοποίηση της προστιθέμενης αξίας των ΤΠΕ στη μαθησιακή διαδικασία, φύλλα εργασίας, κλπ.)
- ΣΤ. Η **αξιολόγηση** (μαθητή και σεναρίου) και οι πιθανές επεκτάσεις του σεναρίου
- Z. **Παρατηρήσεις** και **οδηγίες** για τους εκπαιδευτικούς, **βιβλιογραφία**

Οι προηγούμενες φάσεις εξαρτώνται σε μεγάλο βαθμό μεταξύ τους και συνήθως δεν αναπτύσσονται με γραμμικό τρόπο παρότι υπάρχει μια λογική ακολουθία στη σειρά υλοποίησης. Για παράδειγμα, ο καθορισμός του διδακτικού αντικείμενου προηγείται της διατύπωσης των στόχων. Το διδακτικό υλικό είναι συνυφασμένο με τους στόχους αλλά και τις δραστηριότητες υλοποίησης του σεναρίου και σε μεγάλο βαθμό αναπτύσσονται παράλληλα. Η ανάδειξη των πρότερων ιδεών και των αναπαραστάσεων προδιαγράφει τμήμα των δραστηριοτήτων υλοποίησης του σεναρίου: πρόκειται για δραστηριότητες που στοχεύουν ενδεχομένως στην ανασκευή των πρότερων ιδεών και την αναδόμηση των αναπαραστάσεων που έχουν οι μαθητές σχετικά με τις προς μελέτη έννοιες.

Δεδομένου ότι το εκπαιδευτικό σενάριο προβλέπει τη λειτουργική ένταξη των ΤΠΕ στην υλοποίηση και την εφαρμογή του, η σχεδίαση πρέπει να προβλέψει και να τεκμηριώσει την χρήση ενός τουλάχιστον υπολογιστικού περιβάλλοντος σε κάποιες από τις φάσεις (με έμφαση σε αυτές που αφορούν την υλοποίηση δραστηριοτήτων μέσα στην τάξη) που περιγράφονται αναλυτικά στη συνέχεια.

A. Το διδακτικό αντικείμενο του εκπαιδευτικού σεναρίου

Στην αρχική φάση καθορίζεται το προς μελέτη διδακτικό αντικείμενο, το περιεχόμενο δηλαδή του μαθήματος, προσδιορίζονται τα βασικά τμήματα του σεναρίου και γίνεται εστίαση στα επιμέρους σημεία του αντικείμενου της μάθησης. Συνεπώς, ορίζεται ο τίτλος και το θέμα του σεναρίου, η τάξη ή οι τάξεις στις οποίες μπορεί να απευθύνεται, οι εμπλεκόμενες γνωστικές περιοχές και η συμβατότητα (ή όχι) με το ισχύον αναλυτικό πρόγραμμα. Καθορίζεται επίσης η ενδεικτική διάρκεια υλοποίησης του σεναρίου στην τάξη.

Από τη στιγμή που γίνεται ο καθορισμός του διδακτικού αντικείμενου είναι απαραίτητο να προσδιοριστούν τα βασικά τμήματα του σεναρίου, να γίνει αναφορά στις προαπαιτούμενες γνώσεις που πρέπει να διαθέτουν (τι πρέπει να ξέρουν) οι μαθητές καθώς και στις πρότερες (προϋπάρχουσες) γνώσεις που διαθέτουν πραγματικά (τι ήδη ξέρουν) και να αιτιολογηθεί σύντομα γιατί το προτεινόμενο σενάριο είναι κατάλληλο για το επίπεδο γνώσεων των μαθητών. Η παρούσα φάση αφορά αποκλειστικά τους εκπαιδευτικούς και δεν αποτελεί αντικείμενο εργασίας με τους μαθητές. Μια συνεκτική ανάπτυξη της φάσης αυτής απαιτεί απαντήσεις στα παρακάτω τουλάχιστον ερωτήματα:

A) Ποια είναι τα επί μέρους τμήματα του διδακτικού σεναρίου πάνω στα οποία πρέπει να εργαστούν οι μαθητές, ώστε να *εισαχθούν* και να *οικοδομήσουν* την προς μελέτη έννοια ή έννοιες; Καλύπτει διάφορες πλευρές και εστιάζει στα επίμαχα και σημαντικά σημεία του αντικείμενου μάθησης;

Β) Λαμβάνει υπόψη του το σενάριο τις *προαπαιτούμενες γνώσεις* που πρέπει να διαθέτουν οι μαθητές; Με ποιους τρόπους αποτιμώνται οι γνώσεις αυτές και πώς το σενάριο τις αξιοποιεί οργανικά στην οικοδόμηση των προς απόκτηση γνώσεων;

Γ) Λαμβάνει – και εάν ναι, πώς – υπόψη του το σενάριο τις *πρότερες (προϋπάρχουσες) γνώσεις* που διαθέτουν οι μαθητές για την υπό μελέτη έννοια; Με ποιους τρόπους αποτιμώνται οι γνώσεις αυτές και πώς το σενάριο τις εντάσσει οργανικά στην οικοδόμηση των προς απόκτηση γνώσεων;

Δ) Είναι κατάλληλο το διδακτικό σενάριο για το *επίπεδο γνώσεων* του μαθητή; (δηλαδή ούτε πολύ απλό, ούτε πολύ σύνθετο, ώστε να μπορεί ο μαθητής να το υλοποιήσει με τη βοήθεια ενδεχομένως του εκπαιδευτικού);

Η φάση αυτή, επί της ουσίας, καθορίζει το περιεχόμενο (θέμα, τίτλος, αντικείμενο) του σεναρίου και τη σύνδεσή του με τις γνώσεις των μαθητών. Επιπροσθέτως, σε συνδυασμό με την επόμενη φάση (φάση Β), περιγράφει συνοπτικά το κυρίως σκεπτικό του σεναρίου: για ποιους λόγους δημιουργήθηκε και ποια διδακτικά προβλήματα θέλει να αντιμετωπίσει.

Επιπρόσθετα, λόγω της ένταξης των ΤΠΕ στο σενάριο, είναι απαραίτητο να προσδιοριστούν στη φάση τα γνωστικά προαπαιτούμενα που αφορούν τα προς χρήση λογισμικά και υπολογιστικά εργαλεία.

Β. Οι αναπαραστάσεις των μαθητών και πιθανές δυσκολίες της σκέψης τους

Στη φάση αυτή γίνεται χρήση της υπάρχουσας σχετικής βιβλιογραφίας⁴ αλλά και της εν γένει εμπειρίας του εκπαιδευτικού ώστε να προσδιοριστούν με ακρίβεια οι πιθανές δυσκολίες της σκέψης του μαθητή σχετικά με το προς μελέτη γνωστικό αντικείμενο. Ειδικότερα είναι απαραίτητο να προσδιοριστούν οι πρότερες ιδέες και οι αναπαραστάσεις που διαθέτουν οι μαθητές της ηλικίας που αφορά το σενάριο για την έννοια ή τις έννοιες που μελετούνται. Παράλληλα προσδιορίζονται πιθανές παρανοήσεις και λάθη που κάνουν οι μαθητές στο εν λόγω γνωστικό αντικείμενο και οργανώνεται ένας αναλυτικός κατάλογος με τις γνωστικές δυσκολίες που αντιμετωπίζουν στο πλαίσιο αυτό οι μαθητές. Τα δεδομένα της φάσης αυτής είναι απαραίτητα αφενός για τον καθορισμό των στόχων του σεναρίου (φάση Γ) και αφετέρου για την ανάπτυξη του διδακτικού υλικού (φάση Δ) και των διδακτικών δραστηριοτήτων του σεναρίου (φάση Ε). Είναι προφανές ότι στις επόμενες φάσεις πρέπει να

⁴ Μεγάλο μέρος της βιβλιογραφίας που καταγράφει τις δυσκολίες που αντιμετωπίζουν οι μαθητές στα επιμέρους γνωστικά αντικείμενα καθώς και οι πρότερες ιδέες και αναπαραστάσεις τους υπάρχει πλέον στο Διαδίκτυο και μπορεί εύκολα να εντοπισθεί με τη χρήση μιας μηχανής αναζήτησης.

επεξηγηθεί πως θα αντιμετωπισθούν διδακτικά όλες οι δυσκολίες της σκέψης του μαθητή που εντοπίζονται στην παρούσα φάση. Μια πλήρης ανάπτυξη της φάσης αυτής απαιτεί να δοθούν απαντήσεις στα παρακάτω τουλάχιστον ερωτήματα:

Α) Λαμβάνει - και εάν ναι, πώς - υπόψη του το διδακτικό σενάριο τις *πρότερες ιδέες* και τις *αναπαραστάσεις* των μαθητών για την υπό μελέτη έννοια; Βασίζεται πάνω στις αναπαραστάσεις αυτές και προτείνει λύσεις μετασχηματισμού τους; Πώς ανιχνεύονται οι πρότερες ιδέες και αναπαραστάσεις των μαθητών; Με ποιες γνωστικές και πρακτικές διαδικασίες υποστηρίζεται ο μετασχηματισμός τους;

Β) Λαμβάνει - και εάν ναι, πώς - υπόψη του το διδακτικό σενάριο τις *πιθανές παρανοήσεις* και τα *ενδεχόμενα λάθη* των μαθητών για την υπό μελέτη έννοια; Πώς ανιχνεύονται οι πιθανές παρανοήσεις και τα λάθη των μαθητών; Με ποιους τρόπους επιδιώκεται η ανασκευή των παρανοήσεων και η υπέρβαση των λαθών των μαθητών;

Γ) Λαμβάνει - και εάν ναι, πώς - υπόψη του το σενάριο τις *γνωστικές δυσκολίες* που αντιμετωπίζουν οι μαθητές για την υπό μελέτη έννοια; Πώς εντοπίζονται οι δυσκολίες αυτές και ποιοι είναι οι δυνατοί τρόποι υπέρβασής τους;

Η φάση αυτή συνδέει το σενάριο με τα ευρήματα της διδακτικής των επιστημών (βλέπε ενότητα 4) αλλά και της εμπειρικής γνώσης του εκπαιδευτικού για τα γνωστικά προβλήματα που αντιμετωπίζουν οι μαθητές σχετικά με τις προς μελέτη έννοιες.

Γ. Οι στόχοι του εκπαιδευτικού σεναρίου

Στη φάση αυτή προσδιορίζονται οι στόχοι του εκπαιδευτικού σεναρίου. Οι στόχοι αυτοί καθορίζονται με βάση δύο συμπληρωματικούς άξονες: α) ως προς το γνωστικό αντικείμενο και τη μαθησιακή διαδικασία και β) ως προς τη χρήση των ΤΠΕ.

α) Οι διδακτικοί και οι μαθησιακοί στόχοι του εκπαιδευτικού σεναρίου καθορίζονται συναρτήσει του προς μελέτη διδακτικού αντικειμένου, όπως ορίζει το πρόγραμμα σπουδών (χωρίς να σημαίνει ότι δεν είναι κάποιες φορές ζητούμενο η υπέρβαση του ισχύοντος προγράμματος σπουδών), και των δυσκολιών που παρουσιάζει η σκέψη των μαθητών σε σχέση με το διδακτικό αντικείμενο. Με βάση τους στόχους αυτούς καθορίζονται στη συνέχεια οι διδακτικές δραστηριότητες του εκπαιδευτικού σεναρίου και το προς ανάπτυξη διδακτικό υλικό καθώς και η χρήση των ΤΠΕ.

Η διατύπωση των στόχων γίνεται ρητά και με ακρίβεια (υποδεικνύοντας στον εκπαιδευτικό τους διδακτικούς του στόχους και στο μαθητή το αντικείμενο της μάθησής του) ενώ οι στόχοι αυτοί μπορεί να είναι

- 1) υψηλού επιπέδου και να αφορούν συγκεκριμένες **ικανότητες**,
- 2) χαμηλού επιπέδου και να αφορούν, στην περίπτωση αυτή, απλές **γνώσεις, δεξιότητες και στάσεις**.

Κάθε διδακτική δραστηριότητα του εκπαιδευτικού σεναρίου υποστηρίζει την επίτευξη ενός ή περισσότερων στόχων του σεναρίου. Είναι απαραίτητο να σχεδιαστούν κατάλληλες διδακτικές δραστηριότητες που να καλύπτουν το σύνολο των στόχων του σεναρίου. Παράλληλα, ένας στόχος μπορεί να αντιμετωπίζεται σε παραπάνω από μία δραστηριότητες του σεναρίου.

β) Οι στόχοι ως προς τη χρήση των ΤΠΕ αναδεικνύουν την ανάγκη της ένταξης της τεχνολογίας (λογισμικά, υπηρεσίες, εργαλεία) σε επιμέρους φάσεις της διδασκαλίας. Οι στόχοι αυτοί σχετίζονται με τα επιμέρους χαρακτηριστικά της χρησιμοποιούμενης τεχνολογίας και των δυνατοτήτων που παρέχει για την επίτευξη των διδακτικών στόχων του σεναρίου. Είναι απαραίτητο να υπάρχουν στόχοι που να αξιοποιούν τις δυνατότητες της τεχνολογίας ώστε να εισάγονται νέες διδακτικές πρακτικές. Ειδικότερη αναφορά πρέπει να γίνει σε στόχους που εκμεταλλεύονται την προστιθέμενη αξία των ΤΠΕ στο πλαίσιο του σεναρίου.

Συνεπώς, η φάση αυτή προσδιορίζει τους στόχους του σεναρίου και προδιαγράφει αναπόδραστα το απαιτούμενο διδακτικό υλικό, τους τρόπους με τους οποίους θα χρησιμοποιηθεί η τεχνολογία και τις προς επίτευξη δραστηριότητες μέσα στην τάξη ώστε να επιτευχθούν οι στόχοι αυτοί.

Οι στόχοι του σεναρίου θα πρέπει να μετασηματίζουν δυναμικά τους στόχους της ενότητας, εντάσσοντας τα ψηφιακά εργαλεία, τα οποία προσφέρουν δυνατότητες που αναφέρθηκαν παραπάνω. Γενικά, όσον αφορά τους μαθητές, οι δραστηριότητες που περιγράφονται στα σεναρία θα πρέπει να αναδεικνύουν τα ποιοτικά χαρακτηριστικά των ψηφιακών εργαλείων. Ο όρος «ποιοτικά» δεν αναφέρεται απλά στην επιλογή ενός άλλου εργαλείου, αλλά στη διαφοροποίηση των δραστηριοτήτων που προκαλεί η χρήση ενός άλλου εργαλείου.

Δ. Το διδακτικό υλικό του εκπαιδευτικού σεναρίου

Στη φάση αυτή περιγράφονται αναλυτικά τα διδακτικά υλικά που απαιτούνται για την υλοποίηση του σεναρίου. Στο διδακτικό υλικό περιλαμβάνεται το έτοιμο διδακτικό υλικό (π.χ. έντυπο υλικό για μαθητές, χάρτες, λογισμικό, κατασκευές, κλπ.) και το συμπληρωματικό υλικό που πρέπει να δημιουργηθεί για το σενάριο. Συνήθως δίνεται έμφαση σε απλά υλικά

καθημερινής ζωής (έτοιμα υλικά ή απλές κατασκευές), τα οποία είναι δυνατόν να βρεθούν ή να δημιουργηθούν εύκολα από τους εκπαιδευτικούς.

Τα φύλλα εργασίας αποτελούν τμήμα του διδακτικού υλικού. Είναι επίσης δυνατόν να γίνει χρήση κατάλληλου εκπαιδευτικού λογισμικού, εάν αυτό είναι διαθέσιμο. Στην περίπτωση αυτή περιγράφονται τα αναγκαία αρχεία του λογισμικού και ο τρόπος χρήσης τους (εάν αυτό απαιτείται από το σενάριο, όπως συνήθως συμβαίνει αφού η χρήση της ψηφιακής τεχνολογίας είναι αυτονόητη στη φάση αυτή).

Τέλος προσδιορίζεται η απαιτούμενη υλικοτεχνική υποδομή, τα εποπτικά μέσα, τα όργανα και τα άλλα εν γένει υλικά (π.χ. υπολογιστές, πίνακες, προβολικό μηχάνημα, κλπ.) και ο τρόπος αξιοποίησής τους στο πλαίσιο του σεναρίου. Το σενάριο συνεπώς πρέπει να προβλέπει πως χρησιμοποιείται το διδακτικό υλικό (φύλλα εργασίας, βιβλίο ή τετράδιο μαθητή, εκπαιδευτικό λογισμικό, κλπ.) από τους μαθητές.

Όπως έχει ήδη τονιστεί, το διδακτικό υλικό υποστηρίζει στενά επιμέρους δραστηριότητες υλοποίησης, οι οποίες θα διεξαχθούν στο πλαίσιο της τάξης. Οι εν γένει μορφές που έχουν οι δραστηριότητες αυτές περιγράφονται στη συνέχεια.

Ε. Οι δραστηριότητες υλοποίησης του εκπαιδευτικού σεναρίου στην τάξη

Η φάση αυτή συνιστά την πιο ουσιαστική φάση σχεδίασης του εκπαιδευτικού σεναρίου αφού κατά τη διάρκειά της περιγράφονται όλες οι απαιτούμενες διαδικασίες που αφορούν τους εκπαιδευτικούς και τους μαθητές μέσα στην τάξη. Είναι η φάση κατά την οποία προκαθορίζονται τόσο οι ενέργειες του εκπαιδευτικού όσο και οι ενέργειες των μαθητών ώστε να επιτευχθούν οι στόχοι του σεναρίου ενώ στο πλαίσιο της αναδεικνύονται ουσιαστικά οι λόγοι για τους οποίους είναι χρήσιμο ή και απαραίτητο να ενταχθούν οι τεχνολογίες στη διαδικασία της διδασκαλίας και της μάθησης.

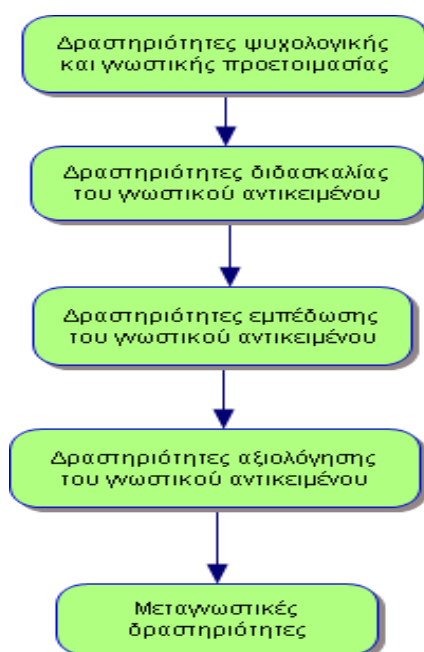
Συνεπώς, για την ανάπτυξη αυτής της φάσης πρέπει να προσδιοριστεί τόσο η θεωρητική και μεθοδολογική προσέγγιση του σεναρίου (θεωρίες μάθησης και διδακτικές καταστάσεις) όσο και οι ακολουθούμενες διδακτικές προσεγγίσεις και στρατηγικές, σε συνάρτηση με τη χρήση των ΤΠΕ και του άλλου διδακτικού υλικού.

Στο πλαίσιο αυτό, κάθε εκπαιδευτικό σενάριο υλοποιείται μέσω ενός συνόλου δραστηριοτήτων μέσα στην τάξη (Σχήμα 2). Οι δραστηριότητες του σεναρίου συνήθως

προτείνονται στους μαθητές με τη μορφή φύλλων εργασίας⁵ και κατά ένα μεγάλο μέρος αφορούν την επίτευξη των στόχων του σεναρίου. Οι δραστηριότητες αυτές χωρίζονται σε πέντε τουλάχιστον διαφορετικές κατηγορίες:

- α) Δραστηριότητες ψυχολογικής και γνωστικής προετοιμασίας για την αποτίμηση της υπάρχουσας γνώσης και την ανίχνευση αναπαραστάσεων και γνωστικών δυσκολιών
- β) Δραστηριότητες διδασκαλίας του γνωστικού αντικείμενου που εισάγουν τις νέες γνώσεις
- γ) Δραστηριότητες εμπέδωσης (κατανόησης δηλαδή και αφομοίωσης των νέων γνώσεων) του γνωστικού αντικείμενου
- δ) Δραστηριότητες αξιολόγησης του γνωστικού αντικείμενου
- ε) Μεταγνωστικές δραστηριότητες

Η αλληλουχία των δραστηριοτήτων αυτών ακολουθεί μια συνηθισμένη δομή σχεδιασμού της διδασκαλίας, η οποία είναι ανεξάρτητη από την ενδεχόμενη χρήση της τεχνολογίας. Στο σημείο αυτό είναι κρίσιμο να προσδιοριστούν οι επιμέρους δραστηριότητες κατά τις οποίες θα γίνει χρήση ενός υπολογιστικού περιβάλλοντος και να αναδειχθούν τα συγκριτικά του πλεονεκτήματα σε σχέση με τις παραδοσιακές διδακτικές τεχνικές.



Σχήμα 2: δραστηριότητες υλοποίησης εκπαιδευτικού σεναρίου

⁵ Ένα εκπαιδευτικό σενάριο περιέχει τουλάχιστον ένα φύλλο εργασίας. Συνήθως όμως περιέχει περισσότερα του ενός φύλλα εργασίας.

Οι δραστηριότητες του σεναρίου καθορίζουν σε μεγάλο βαθμό τη μορφή και το περιεχόμενο των φύλλων εργασίας που θα περιέχει το εκπαιδευτικό σενάριο και θα δοθούν στους μαθητές για υλοποίηση μέσα στην τάξη. Κάθε δραστηριότητα χρησιμοποιεί μία ή περισσότερες *διδακτικές στρατηγικές*⁶ ή *διδακτικές τεχνικές*.

α) Δραστηριότητες ψυχολογικής και γνωστικής προετοιμασίας

Οι αρχικές δραστηριότητες του εκπαιδευτικού σεναρίου σχετίζονται με την ψυχολογική και τη γνωστική προετοιμασία και αφορούν τη διαμόρφωση κατάλληλου συναισθηματικού κλίματος στην τάξη, τη διαμόρφωση κινήτρου για το μάθημα, την ενημέρωση για τον σκοπό και τους στόχους του μαθήματος καθώς και την αποτίμηση της υπάρχουσας γνώσης, την ανίχνευση των γνωστικών δυσκολιών και των αναπαραστάσεων των μαθητών. Είναι σκόπιμο να γίνει κατάλληλη αναφορά και στις τεχνολογίες που θα χρησιμοποιηθούν κατά την υλοποίηση του σεναρίου στην τάξη.

Οι δραστηριότητες που σχετίζονται με την αποτίμηση της προαπαιτούμενης και της προϋπάρχουσας γνώσης καθώς και οι δραστηριότητες ανίχνευσης των γνωστικών δυσκολιών των μαθητών που αφορούν ιδέες, αντιλήψεις και αναπαραστάσεις μπορούν να διεξαχθούν με τη χρήση κατάλληλων διδακτικών στρατηγικών όπως είναι οι ερωταποκρίσεις (ζητάμε ορισμό, θέτουμε ένα ερώτημα), ο καταϊγισμός ιδεών, η δημιουργία ή ο σχολιασμός σχεδίων και η εννοιολογική χαρτογράφηση. Στο πλαίσιο αυτό ακολουθούνται συνήθως τα ακόλουθα βήματα:

- Διαμόρφωση κατάλληλου συναισθηματικού κλίματος και κλίματος ασφάλειας για το μαθητή (Διατυπώστε ένα εισαγωγικό σχόλιο που θα χρησιμοποιήσετε)
- Διαμόρφωση κατάλληλης αφόρμησης για το μάθημα (Διατυπώστε ένα εισαγωγικό σχόλιο ώστε να προκαλέσετε το ενδιαφέρον του μαθητή για το μάθημα)
- Ενημέρωση των μαθητών για το τι θα επακολουθήσει (Με πολύ λίγες λέξεις)
- Ενημέρωση των μαθητών για τον σκοπό και τους στόχους του μαθήματος (Μετά το τέλος του μαθήματος θα πρέπει να είστε σε θέση να...)
- Διερεύνηση προϋπάρχουσας και προαπαιτούμενης γνώσης (Τι ξέρετε για...)

⁶ Στην υπάρχουσα βιβλιογραφία ο όρος *διδακτική τεχνική* χρησιμοποιείται συνήθως εναλλακτικά του όρου *διδακτική στρατηγική*. Στο παρόν κείμενο θα χρησιμοποιούμε τον όρο *διδακτική στρατηγική*. Με τον όρο αυτό θα εννοούμε μια τεχνική, βασισμένη συνήθως σε αρχές μιας παιδαγωγικής θεωρίας ή μιας θεωρίας μάθησης, μέσω της οποίας επιδιώκεται επίτευξη ενός μαθησιακού αποτελέσματος.

- Διερεύνηση ιδεών, αντιλήψεων, αναπαραστάσεων (Τι έχετε ακούσει ή διαβάσει για... Τι νομίζετε για...)

Στη φάση αυτή είναι δυνατόν να γίνει χρήση κατάλληλων εκπαιδευτικών λογισμικών, τα οποία επιτρέπουν την ανίχνευση των πρότερων γνώσεων και ιδεών των μαθητών (π.χ. επεξεργαστές κειμένου, λογισμικά εννοιολογικής χαρτογράφησης, ζωγραφική, κλπ.).

β) Δραστηριότητες διδασκαλίας του γνωστικού αντικειμένου

Η φάση αυτή καταλαμβάνει συνήθως το μεγαλύτερο μέρος του εκπαιδευτικού σεναρίου αφού στο πλαίσιο της εισάγονται οι προς απόκτηση γνώσεις και διεξάγονται οι περισσότερες δραστηριότητες πρόσκτησης και οικοδόμησης των νέων εννοιών. Σημαντική επίσης σε αυτή την ενότητα είναι η χρονική αλληλουχία των ενεργειών εκπαιδευτικού και μαθητών. Σε συνδυασμό με τη φάση εμπέδωσης των γνώσεων (βλέπε στη συνέχεια) αποτελούν τα στάδια εκείνα όπου ενισχύονται οι προϋπάρχουσες γνώσεις, ανασκευάζονται οι λανθασμένες αντιλήψεις και οι αρχικές ιδέες των μαθητών, αναδομούνται οι αναπαραστάσεις και δημιουργείται το κατάλληλο πλαίσιο για την εννοιολογική αλλαγή και την οικοδόμηση των νέων γνώσεων.

Στη φάση αυτή απαιτείται οργανική ένταξη και ενσωμάτωση του ή των χρησιμοποιούμενων υπολογιστικών περιβαλλόντων. Συνεπώς, είναι απαραίτητο να τεκμηριωθούν επαρκώς οι λόγοι για τους οποίους απαιτείται η χρήση ενός υπολογιστικού περιβάλλοντος για την υλοποίηση του εκπαιδευτικού σεναρίου και ειδικότερα να τονισθεί η προστιθέμενη αξία και γενικά τα πλεονεκτήματα που προσθέτει το περιβάλλον αυτό στην επίτευξη του σκοπού και των στόχων του σεναρίου.

Η τεκμηρίωση της προστιθέμενης αξίας που παρέχει το υπολογιστικό περιβάλλον μπορεί να γίνει με πολλαπλούς τρόπους. Συνήθως, δίνεται έμφαση στις δυνατότητες για δράση που προσφέρουν τα προτεινόμενα υπολογιστικά εργαλεία ή περιβάλλοντα. Είναι σκόπιμο να γίνεται σύγκριση των εν λόγω δυνατοτήτων με τις εν γένει δυνατότητες που έχει στη διάθεσή του η εκπαιδευτικός ή ο μαθητής σε συμβατικό σχολικό περιβάλλον (χωρίς δηλαδή την χρήση των ΤΠΕ). Το πλαίσιο χρήσης που διαμορφώνουν τα υπολογιστικά εργαλεία διακρίνεται σε ποσοτικό ή και σε ποιοτικό επίπεδο από το παραδοσιακό πλαίσιο, στο οποίο δεν υπάρχουν υπολογιστές και με ποιους τρόπους;

Οι δραστηριότητες που απαιτούνται στη φάση αυτή πρέπει να καλύπτουν τα παρακάτω τουλάχιστον ερωτήματα:

1. Τι τύπου *διδακτικές στρατηγικές* χρησιμοποιεί το εκπαιδευτικό σενάριο; Ποιες είναι οι απαιτούμενες ενέργειες από τον εκπαιδευτικό για την υλοποίησή τους;
2. Τι τύπου *διδακτικές καταστάσεις*⁷ ευνοεί το εκπαιδευτικό σενάριο (όπως ατομικές ή συλλογικές, κλειστές ή ανοικτές, επεκτάσιμες). Ποιες είναι οι απαιτούμενες ενέργειες από τον εκπαιδευτικό για την υλοποίησή τους;
3. Τι τύπου *διδακτικές βοήθειες*⁸ προτείνει το σενάριο; Ποιες είναι οι απαιτούμενες ενέργειες από τον εκπαιδευτικό για την υλοποίησή τους; Πώς εξελίσσεται η διαδικασία με τις παρεμβάσεις του εκπαιδευτικού;
4. Προτείνει γνωστικού τύπου συγκρούσεις και ποιος είναι ο ρόλος του εκπαιδευτικού και του χρησιμοποιούμενου εκπαιδευτικού υλικού ή λογισμικού σε αυτή τη διαδικασία;
5. Πώς οργανώνονται οι *αλληλεπιδράσεις* ανάμεσα στους μαθητές, το χρησιμοποιούμενο υλικό και τον εκπαιδευτικό; Πώς ευνοείται η ουσιαστική συνεργασία ανάμεσα σε μαθητές και εκπαιδευτικό;
6. Το σενάριο ευνοεί τον πειραματισμό και επιτρέπει στο μαθητή να διερευνήσει και να ανακαλύψει τη γνώση;

γ) Δραστηριότητες εμπέδωσης του γνωστικού αντικειμένου

Οι δραστηριότητες *εμπέδωσης* (της κατανόησης δηλαδή και της αφομοίωσης των νέων γνώσεων) χρησιμοποιούν συνήθως παρεμφερείς διδακτικές στρατηγικές με τις δραστηριότητες διδασκαλίας του γνωστικού αντικειμένου. Συνήθως οι δραστηριότητες εμπέδωσης λαμβάνουν χώρα μέσω ερωταποκρίσεων, πρακτικών επίλυσης προβλημάτων και εφαρμογής των γνώσεων που έχουν αποκτηθεί σε συγκεκριμένες καταστάσεις. Το υπολογιστικό περιβάλλον που εμπλέκεται στο σενάριο μπορεί να παίξει καθοριστικό ρόλο και στη φάση αυτή, οπότε είναι απαραίτητη η τεκμηρίωση της χρήσης του. Τα ερωτήματα που τίθενται στην προηγούμενη φάση (δραστηριότητες διδασκαλίας του γνωστικού αντικειμένου) σχετικά με τις διδακτικές στρατηγικές, τις διδακτικές καταστάσεις, τις

⁷ Η διδακτική κατάσταση ορίζεται ως το σύνολο των οργανωμένων ενεργειών του εκπαιδευτικού που αφορούν τις σχέσεις ανάμεσα σε ένα υποκείμενο που μαθαίνει και σε ένα υποκείμενο που διδάσκει και το περιβάλλον που κινητοποιεί ο εκπαιδευτικός ώστε ο μαθητής να αποκτήσει ή να οικοδομήσει μια συγκεκριμένη γνώση.

⁸ Η διδακτική βοήθεια αφορά την υποστήριξη ή την καθοδήγηση που προσφέρει ο εκπαιδευτικός άλλοτε ρητά και άλλοτε άρρητα στους μαθητές. Μπορεί να είναι υποστηρικτική, συνεργατική ή καθοδηγητική και βασίζεται στον προφορικό λόγο του εκπαιδευτικού αλλά και στο χρησιμοποιούμενο διδακτικό υλικό (σχήματα, κατασκευές, λογισμικό, κλπ.). Διδακτική βοήθεια μπορεί επίσης να προσφέρει κάποιος μαθητής σε κάποιον άλλο μαθητή στο πλαίσιο μιας συνεργατικής δραστηριότητας.

διδακτικές βοήθειες, την οργάνωση των αλληλεπιδράσεων, κλπ. αφορούν προφανώς και τη φάση αυτή. Επιπρόσθετες δραστηριότητες που απαιτούνται στη φάση αυτή πρέπει να καλύπτουν τα παρακάτω τουλάχιστον ερωτήματα:

1. Ποιες ερωτήσεις εμπέδωσης θέτει το σενάριο ώστε να ενθαρρύνεται η κατασκευή της γνώσης από τους μαθητές λαμβάνοντας υπόψη τις δυσκολίες που αντιμετωπίζουν για την προς μελέτη έννοια;
2. Ποιες δραστηριότητες εξάσκησης και πρακτικής προτείνει το σενάριο που αφορούν άμεσα τις γνώσεις που πρέπει να εμπέδωθούν;
3. Τι τύπου προβληματικές καταστάσεις προτείνονται στους μαθητές μέσω του σεναρίου ώστε να υποστηριχθεί η εμπέδωση των γνώσεων που έχουν αποκτηθεί στο πλαίσιο του;

δ) Δραστηριότητες αξιολόγησης του γνωστικού αντικειμένου

Οι δραστηριότητες αυτές, παρότι αποτελούν εγγενές τμήμα της υλοποίησης του εκπαιδευτικού σεναρίου, μελετώνται στην επόμενη φάση (φάση ΣΤ) που άπτεται της γενικότερης αξιολόγησης στο πλαίσιο του σεναρίου.

ε) Μεταγνωστικές δραστηριότητες

Οι μεταγνωστικές δραστηριότητες συνιστούν εγγενές τμήμα του σεναρίου και λαμβάνουν χώρα τόσο στην τάξη όσο και εκτός τάξης. Γίνεται συνήθως αντιπαραβολή και σύγκριση των γνώσεων που αποκτήθηκαν με τις αρχικές ιδέες και αναπαραστάσεις και προτείνεται δουλειά για το σπίτι, εάν αυτό προβλέπεται από το πρόγραμμα σπουδών. Ο όρος «μεταγνώση» αναφέρεται στη γνώση που διαθέτουμε σχετικά με τη δική μας γνωστική διαδικασία (και αφορά την αντίληψη, την προσοχή, τη μνήμη, τη σκέψη, τη γλώσσα και τη μάθηση), στο αποτέλεσμα της και σε οτιδήποτε σχετίζεται με αυτά. Η μεταγνώση αφορά στη μάθηση της μάθησης (ή με άλλα λόγια στην επίγνωση του τρόπου με τον οποίο μαθαίνουμε). Οι μεταγνωστικές δραστηριότητες διεξάγονται συνήθως προς το τέλος ενός εκπαιδευτικού σεναρίου. Στις δραστηριότητες αυτές η εκπαιδευτικός καλεί τους μαθητές να επιστρέψουν και διατυπώσουν την αρχική «δική τους» εναλλακτική ιδέα ή αναπαράσταση (αν χρειαστεί τους την παρουσιάζει η ίδια), να την αντιπαραβάλλουν με την προτεινόμενη από το πρόγραμμα σπουδών, να αναστοχαστούν πάνω σε αυτές και να συζητήσουν ή να καταγράψουν τα υπέρ και τα κατά των δύο διαφορετικών ιδεών. Η σχεδίαση και υλοποίηση μεταγνωστικών δεξιοτήτων που υποστηρίζονται από κατάλληλα υπολογιστικά περιβάλλοντα είναι πλέον συνήθης πρακτική. Είναι συνεπώς απαραίτητο να τεκμηριωθεί ο τρόπος με τον

οποίο το χρησιμοποιούμενο από το σενάριο υπολογιστικό περιβάλλον συμβάλει στη φάση αυτή. Συνήθως οι μεταγνωστικές δραστηριότητες αφορούν:

- Σύνοψη του μαθήματος και των νέων γνώσεων που αποκτήθηκαν (συνήθως δημιουργείται ένα διάγραμμα σύνοψης ή ένας εννοιολογικός χάρτης).
- Αντιπαραβολή των αρχικών ιδεών και αναπαραστάσεων των μαθητών με τις νέες γνώσεις που αποκτήθηκαν στο μάθημα (τι πιστεύαμε αρχικά, τι νέο έχουμε μάθει, πως έχουν αλλάξει οι ιδέες και οι απόψεις μας ...). Μπορεί να υλοποιηθεί με τη σύγκριση αρχικών και τελικών εννοιολογικών χαρτών.
- Δουλειά για το σπίτι (να περιγραφεί σαφώς, όταν απαιτείται από το σενάριο)
- Μεταγνωστική αξιολόγηση (τι κερδίσατε από αυτό το μάθημα;)

στ) Η αξιολόγηση (μαθητή και εκπαιδευτικού σεναρίου)

Η αξιολόγηση αφορά τόσο τη μαθησιακή πορεία των μαθητών όσο και το ίδιο το σενάριο. Ένα ολοκληρωμένο εκπαιδευτικό σενάριο εμπεριέχει συγκεκριμένες δραστηριότητες σχετικές με την αξιολόγηση των μαθητών. Οι δραστηριότητες αυτές αποτελούν εγγενές τμήμα του σεναρίου και χρησιμοποιούνται ώστε να εκτιμήσουμε την αποτελεσματικότητα των δραστηριοτήτων που προηγήθηκαν. Πρόκειται συχνά για δραστηριότητες που προσφέρουν εκ νέου την ευκαιρία σε μαθητές να προσεγγίσουν ζητήματα στα οποία ενδέχεται να έχουν ακόμα γνωστικές ανάγκες.

Για κάθε στόχο που έχει τεθεί στη φάση Γ του σεναρίου είναι σκόπιμο να δημιουργηθεί ένα κριτήριο (όπως ερώτηση, άσκηση, πρόβλημα, εννοιολογικός χάρτης, κλπ.) με το οποίο θα ελέγχεται η επίτευξή του.

Η αξιολόγηση περιλαμβάνει συνήθως

1. Ασκήσεις σωστού – λάθους, πολλαπλών επιλογών, συμπλήρωσης κενών, κλπ.
2. Ερωτήσεις αξιολόγησης (ανοικτού τύπου) που θέτει το σενάριο ώστε να διερευνηθεί η κατανόηση της υπό μελέτη έννοιας από τους μαθητές.
3. Δραστηριότητες σχεδίασης (π.χ. κάνουν μία ζωγραφιά, δημιουργούν ένα διάγραμμα ροής),
4. Δραστηριότητες εννοιολογικής χαρτογράφησης (δημιουργούν έναν νοητικό χάρτη),
5. Δραστηριότητες επίλυσης προβλημάτων,
6. Δραστηριότητες κατασκευής (π.χ. δημιουργούν μια κατασκευή).

Σε όλες τις παραπάνω περιπτώσεις είναι επιθυμητή η ενσωμάτωση υπολογιστικών περιβαλλόντων, τα οποία μπορούν να χρησιμοποιηθούν ως εργαλεία λιγότερο ή περισσότερο αποτελεσματικής αξιολόγησης.

Η αξιολόγηση του σεναρίου εκλαμβάνει συνήθως δύο μορφές: α) διαμορφωτική αξιολόγηση κατά τη διάρκεια εφαρμογής του σεναρίου ώστε να αντιμετωπισθούν πιθανά προβλήματα που ανακύπτουν από την εφαρμογή του και β) τελική αξιολόγηση που σχετίζεται με την επίτευξη των στόχων του σεναρίου, τη διαδικασία υλοποίησής του, την αποτελεσματική χρήση των χρησιμοποιούμενων εργαλείων, κλπ.

Z. Οδηγίες – παρατηρήσεις, βιβλιογραφία

Στη φάση αυτή παρέχονται τυχόν οδηγίες και παρατηρήσεις που πρέπει να λάβει υπόψη του η εκπαιδευτικός και αφορούν την ορθή διεξαγωγή του σεναρίου. Στο πλαίσιο αυτό, καταγράφονται όλα τα απαραίτητα σχόλια, παρατηρήσεις και οδηγίες που χρειάζεται μια εκπαιδευτικός ώστε να μπορέσει να πραγματοποιήσει το εκπαιδευτικό σενάριο σε περιβάλλον πραγματικής τάξης. Τέλος, είναι απαραίτητο να δοθούν όλα τα στοιχεία της βιβλιογραφίας που χρησιμοποιήθηκε για την υλοποίηση του σεναρίου.

Εναλλακτικό μοντέλο σχεδίασης εκπαιδευτικών σεναρίων με αφετηρία και βάση τις «Ανάγκες των μαθητών»

Το μοντέλο που προτείνεται αποτελεί ένα ενδεικτικό εναλλακτικό μοντέλο σχεδίασης εκπαιδευτικών σεναρίων με αφετηρία και βάση τις ανάγκες των μαθητών. Το γεγονός ότι η σχεδίαση του εκπαιδευτικού σεναρίου ξεκινάει από τις ανάγκες των μαθητών, το καθιστά περισσότερο μαθητοκεντρικό και λιγότερο γνωσιοκεντρικό από αυτό που έχει ήδη προταθεί στην προηγούμενη ενότητα. Είναι όμως ανάγκη να τεκμηριωθεί, να εφαρμοσθεί, να αξιολογηθεί και να αναπτυχθεί περαιτέρω ώστε να αποτελέσει μια ολοκληρωμένη πρόταση διδακτικού σχεδιασμού. Στη συνέχεια περιγράφονται οι φάσεις του προτεινόμενου μοντέλου οι οποίες βρίσκονται σε στενή σχέση και αλληλεπίδραση.

A. Ανάγκες των μαθητών που αφορούν:

1. στην κοινωνική τους συμβίωση και λειτουργία (κατανόηση, προσαρμογή, σύγκρουση, αναθεώρηση, διαμόρφωση, δημιουργία κτλ)
2. στη σχέση τους με το φυσικό και τεχνολογικό περιβάλλον (κατανόηση, αξιοποίηση/χρήση, πρόβλεψη, δημιουργία, προστασία)

3. στη σχέση τους με τους ίδιους τους εαυτούς τους (φυσιολογικές ανάγκες, ανάγκες ασφάλειας, κοινωνικές ανάγκες, ανάγκες αναγνώρισης, ανάγκες αυτοολοκλήρωσης)

Με βάση την κάλυψη κάποιας ή κάποιων από τις παραπάνω ανάγκες πραγματοποιείται ο

Β. «Καθορισμός των διδακτικών στόχων»

και με βάση τους διδακτικούς στόχους πραγματοποιείται ο

Γ. «Προσδιορισμός του διδακτικού αντικειμένου».

Στη συνέχεια πραγματοποιείται η

Δ. Ανίχνευση πρότερων γνώσεων & αναπαραστάσεων, ικανοτήτων και στάσεων, η

Ε. Δημιουργία Διδακτικού Υλικού και Δραστηριοτήτων Σεναρίου και η

ΣΤ. Αξιολόγηση

με τον ίδιο τρόπο με τον οποίο πραγματοποιείται στο μοντέλο που αναπτύχθηκε στην προηγούμενη ενότητα.

ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ ΚΑΙ ΕΡΓΑΣΙΕΣ

Δραστηριότητα 1η	Αναπτύξτε μία εκπαιδευτική δραστηριότητα δύο (2) διδακτικών ωρών που να αφορά στην εφαρμογή και στην αξιολόγηση του μοντέλου ΓΤΠΠ σε ένα μάθημα Πληροφορικής. Δηλαδή σχεδιάστε ένα μάθημα 2 (διδακτικών) ωρών και προσπαθήστε να προσδιορίσετε τα στοιχεία ΓΤΠΠ που απαιτούνται για τη διδασκαλία του.
Δραστηριότητα 1*	Μελετήστε το θεωρητικό μέρος και επισημάνετε τους ορισμούς βασικών εννοιών: Θεωρία ΓΤΠΠ Διδακτικά σενάρια.
Δραστηριότητα 2 ^η	Θεωρείστε ένα από τα σενάρια που έχετε ήδη χρησιμοποιήσει (σε κάποιο περασμένο μάθημα) και δοκιμάστε να εντοπίσετε τα σημεία που αναφέρονται στη θεωρητική περιγραφή
Δραστηριότητα 3η	Αναπτύξτε μια εκπαιδευτική δραστηριότητα δυο ωρών (3) ωρών που να αφορά στην ανάπτυξη, παρουσίαση και αξιολόγηση - ανά

μικρή ομάδα εργασίας- ενός αυθεντικού εκπαιδευτικού σεναρίου,
το οποίο χρησιμοποιεί ΤΠΕ στα μαθήματα της Πληροφορικής
ακολουθώντας ένα συγκεκριμένο μοντέλο ανάπτυξης
εκπαιδευτικών σεναρίων

ΕΡΩΤΗΣΕΙΣ ΚΛΕΙΣΤΟΥ ΤΥΠΟΥ

- (1) Ως διδακτικό σενάριο, θεωρούμε την περιγραφή μιας διδασκαλίας με εστιασμένο γνωστικό αντικείμενο, συγκεκριμένους εκπαιδευτικούς στόχους, διδακτικές αρχές και πρακτικές και προσδιορισμένους διδακτικούς πόρους.

ΣΩΣΤΟ ΛΑΘΟΣ

- (2) Προκειμένου να διδαχθεί μια ενότητα η οποία (γνωρίζουμε ότι) παρουσιάζει αρκετές δυσκολίες για τους μαθητές, μπορούμε να επιλέξουμε ανάμεσα από διαφορετικούς τύπους λογισμικών έτσι ώστε να οργανώσουμε ένα κατάλληλο διδακτικό σενάριο (ενδεχομένως με διάρκεια μεγαλύτερη από 1 διδακτική ώρα).

Η ενότητα αυτή, για μια αποτελεσματική της διδασκαλία, απαιτεί υπολογιστικά εργαλεία με τα ακόλουθα χαρακτηριστικά:

- (α) ένα περιβάλλον το οποίο να επιτρέπει την εκτέλεση εύκολων, αλλά μεγάλου πλήθους αριθμητικών πράξεων
- (β) ένα περιβάλλον το οποίο να επιτρέπει τη σύνταξη γραπτών αναφορών οι οποίες να είναι διαμοιράσιμες μεταξύ πολλών χρηστών για περαιτέρω παράλληλη επεξεργασία
- (γ) ένα περιβάλλον στο οποίο να είναι δυνατή παρακολούθηση της εξέλιξης ενός φαινομένου, υπό τον έλεγχο του χρήστη (μαθητή ή μαθήτριας).

Ποια από τα λογισμικά που ακολουθούν, σύμφωνα με τις παραπάνω πληροφορίες, είναι τα καταλληλότερα για την οργάνωση μιας αποτελεσματικής διδασκαλίας:

- (1) λογισμικό επεξεργασίας κειμένου ή κειμενογράφου
- (2) λογισμικό πρακτικής και εκγύμνασης (drill and practice)
- (3) υπολογιστικό φύλλο
- (4) λογισμικό που να επιτρέπει την ψηφιακή μεταφορά αρχείων (όπως την επισύναψη σε ηλεκτρονικό μήνυμα)
- (5) λογισμικό καθοδήγησης και μάθησης σχετικό με την προς διδασκαλία έννοια
- (6) λογισμικό προσομοίωσης σχετικό με την προς διδασκαλία έννοια

Καταλληλότερα είναι τα λογισμικά:

(1) τα 1,2,3,4

(2) τα 2,3,4,6

(3) τα 2,4,5,6

(4) τα 1,3,4,6

(5) τα 1,3,4,5

ΧΡΗΣΗ ΕΞΩΤΕΡΙΚΩΝ ΠΗΓΩΝ

Δεν αναφέρονται

ΒΙΒΛΙΟΓΡΑΦΙΑ

Mishra, P., & Koehler, M. J. (2006). [Technological Pedagogical Content Knowledge: A new framework for teacher knowledge](#). *Teachers College Record* 108 (6), 1017-1054).

Για μια ανάλογη περιγραφή διδακτικών σεναρίων για τα Μαθηματικά: Dagdidelis, V., & I. Papadopoulos 2010. Didactic Scenarios and ICT: A Good Practice Guide. Στο M. D. Lytras, et al. (Eds.), *Technology Enhanced Learning. Quality of Teaching and Educational Reform. First International Conference, TECH EDUCATION 2010. Athens, Greece, May 19-21*, 117-123. Berlin Heidelberg: Springer.

Επίσης το κείμενο:

http://www.greeklanguage.gr/sites/default/files/digital_school/3.1.3ancient_greek_polkas-touloumis_0.pdf

αναπτύσσει προτάσεις για μια εισαγωγή στα διδακτικά σενάρια στα γλωσσικά μαθήματα.

ΔΙΚΤΥΟΓΡΑΦΙΑ

Για μια εισαγωγή στη θεωρία των Mishra και Keller, μπορεί κανείς να συμβουλευτεί τη Wikipedia: http://en.wikipedia.org/wiki/Technological_Pedagogical_Content_Knowledge (τελευταία επίσκεψη 6 Οκτωβρίου 2012).

Παρεμπιπτόντως, είναι ενδιαφέρουσα η ιστορία του WIKI που ήταν αφιερωμένο στη θεωρία του TPACK (περιγράφει ο Koehler):

“The TPACK wiki is dead and gone. Simply put, it was too much effort to be constantly undoing what the spammers were doing to the site. In the last month alone, 400 spam pages were created, the front page was consistently defaced, and 900 users were created (only 2 legitimately edited the site).

That said, the idea of a WIKI is appealing. Instead of opening up the site to editing for all comers, this site will restrict editing to those who are interested in TPACK...”

ΙΣΤΟΧΩΡΟΙ

[Στον ιστοχώρο:](#)

<http://www.tpck.org/>

[όπως και στον:](#)

<http://punya.educ.msu.edu/2008/01/12/mishra-koehler-2006/>

υπάρχουν πολλές αναφορές στη θεωρία των Mishra και Koehler για τη Γνώση Τεχνολογίας, Παιδαγωγικής, Περιεχομένου (τελευταία επίσκεψη 6 Οκτωβρίου 2012).

Ενότητα ΘΕΩΡΙΑ ΔΙΔΑΚΤΙΚΗΣ ΤΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ 3

Διδακτική αντικειμενοστρεφούς προγραμματισμού

1. ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΕΠΙΜΟΡΦΩΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Η διδασκαλία του αντικειμενοστρεφούς προγραμματισμού, όπως προκύπτει από τη διεθνή βιβλιογραφία, συνοδεύεται από ποικίλες δυσκολίες. Ένα από τα σημαντικότερα προβλήματα έγκειται στο γεγονός ότι η διδασκαλία του αντικειμενοστρεφούς προγραμματισμού δεν πραγματοποιείται έχοντας ως βασικό στόχο την ανάδειξη των πλεονεκτημάτων της συγκεκριμένης τεχνικής προγραμματισμού, αλλά την παρουσίαση κατά κύριο λόγο των βασικών προγραμματιστικών εννοιών (δομή ακολουθίας, επιλογής και επανάληψης). Το αποτέλεσμα είναι να μην γίνονται κατανοητές οι βασικές αντικειμενοστρεφείς δομές και στην ουσία οι μαθητές να αναπτύσσουν προγράμματα που βασίζονται στην τεχνική του δομημένου προγραμματισμού χρησιμοποιώντας μια αντικειμενοστρεφή γλώσσα.

Το παρόν επιμορφωτικό σενάριο ασχολείται με το ζήτημα της ανάδειξης των στοιχείων εκείνων που διαδραματίζουν καθοριστικό ρόλο στην αποτελεσματική οργάνωση της διδασκαλίας του αντικειμενοστρεφούς προγραμματισμού σε μαθητές. Διαπραγματεύονται ζητήματα, όπως:

- οι βασικές έννοιες του αντικειμενοστρεφούς προγραμματισμού και η σημασία τους,
- οι καταγεγραμμένες δυσκολίες και παρανοήσεις,
- γνωστά διδακτικά προβλήματα,
- εκπαιδευτικά περιβάλλοντα και μικρόκοσμοι προγραμματισμού.

2. ΕΚΤΙΜΩΜΕΝΗ ΔΙΑΡΚΕΙΑ ΕΠΙΜΟΡΦΩΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Προβλέπεται να διαρκέσει συνολικά 3-4 διδακτικές ώρες.

3. ΣΚΟΠΟΙ ΚΑΙ ΣΤΟΧΟΙ ΤΟΥ ΕΠΙΜΟΡΦΩΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Οι επιμορφούμενοι πρέπει:

- Να γνωρίζουν τις βασικές έννοιες του αντικειμενοστρεφούς προγραμματισμού και να είναι σε θέση να οργανώνουν τη διδασκαλία του με τέτοιο τρόπο ώστε να αναδεικνύεται η σημασία και ο ρόλος τους.
- Να γνωρίζουν τις δυσκολίες/παρανοήσεις που συνοδεύουν την εκμάθηση των βασικών αντικειμενοστρεφών εννοιών και να είναι σε θέση να εντοπίζουν και να αντιμετωπίζουν την εμφάνισή τους στη διδακτική πράξη.
- Να είναι σε θέση να σχεδιάζουν κατάλληλο εκπαιδευτικό υλικό (ερωτήσεις, δραστηριότητες) λαμβάνοντας υπόψη τις προαναφερθείσες δυσκολίες και παρανοήσεις και να το αξιοποιούν κατάλληλα στη διδακτική πράξη.
- Να γνωρίζουν τα καταγεγραμμένα διδακτικά προβλήματα και να είναι σε θέση να οργανώνουν κατάλληλα τη διδασκαλία του αντικειμενοστρεφούς προγραμματισμού ανάλογα με το πλαίσιο στο οποίο εντάσσεται.
- Να γνωρίζουν τις γενικές δυσκολίες που συνοδεύουν ένα μάθημα εισαγωγής στον προγραμματισμό και τα πλεονεκτήματα που παρέχει η χρήση προγραμματιστικών μικρόκοσμων.
- Να γνωρίζουν τους προγραμματιστικούς μικρόκοσμους που υπάρχουν διαθέσιμοι για τη διδασκαλία του αντικειμενοστρεφούς προγραμματισμού.
- Να είναι σε θέση να επιλέγουν τον καταλληλότερο για κάθε περίπτωση μικρόκοσμο και να αξιοποιούν κατάλληλα τις δυνατότητές του στη διδακτική πράξη.

4. ΕΠΙΣΤΗΜΟΛΟΓΙΚΗ ΠΡΟΣΕΓΓΙΣΗ ΚΑΙ ΕΝΝΟΙΟΛΟΓΙΚΗ ΑΝΑΛΥΣΗ

Στόχοι διδασκαλίας αντικειμενοστρεφούς προγρ/μού

Οι εκπαιδευτικοί πρέπει να οργανώνουν τη διδασκαλία του αντικειμενοστρεφούς προγραμματισμού με τέτοιο τρόπο ώστε να δίνεται έμφαση στην κατανόηση των βασικών αντικειμενοστρεφών εννοιών και της αντικειμενοστρεφούς τεχνικής ανάπτυξης προγραμμάτων και όχι απλά στην ανάπτυξη προγραμμάτων με μια αντικειμενοστρεφή γλώσσα. Στόχος δεν αποτελεί η εκμάθηση της σύνταξης μιας αντικειμενοστρεφούς γλώσσας προγραμματισμού, αλλά των εννοιών του αντικειμενοστρεφούς προγραμματισμού.

Οι εκπαιδευτικοί πρέπει να μπορούν να σχεδιάζουν και να εφαρμόζουν εκπαιδευτικές δραστηριότητες που θα συντελούν στην ανάπτυξη ικανοτήτων επίλυσης προβλημάτων με την αντικειμενοστρεφή τεχνική. Ένα αντικειμενοστρεφές πρόγραμμα δημιουργεί στην ουσία στον υπολογιστή ένα μοντέλο ενός συστήματος του κόσμου. Κεντρικό ρόλο στην επίλυση ενός προβλήματος παίζει η μοντελοποίησή του. Ο εκπαιδευτικός πρέπει να δώσει έμφαση

στη διαδικασία εντοπισμού των οντοτήτων (αντικειμένων) από τα οποία είναι δομημένο το μοντέλο του συστήματος που θα αναπαρασταθεί με το πρόγραμμα που θα αναπτυχθεί.

Στην περίπτωση που οι μαθητές έχουν προηγούμενη εμπειρία με μια διαδικαστική γλώσσα προγραμματισμού (ή ψευδογλώσσα) είναι απαραίτητο ο εκπαιδευτικός να τονίσει από την αρχή τις διαφορετικές αρχές που διέπουν τη σχεδίαση της λύσης ενός προβλήματος με την τεχνική του δομημένου και του αντικειμενοστρεφούς προγραμματισμού.

Βασικές έννοιες

Οι δύο βασικότερες έννοιες του αντικειμενοστρεφούς προγραμματισμού είναι η έννοια της κλάσης και του αντικειμένου, οι οποίες μάλιστα σύμφωνα με τη διεθνή βιβλιογραφία πολλές φορές συγχέονται. Η διδασκαλία της συγκεκριμένης τεχνικής προγραμματισμού πρέπει να επικεντρωθεί στις παρακάτω έννοιες:

- Η αναπαράσταση των οντοτήτων/τμημάτων ενός συστήματος του κόσμου για το οποίο δημιουργούμε ένα μοντέλο στον υπολογιστή γίνεται με τα **αντικείμενα**.
- Τα αντικείμενα ανάλογα με τις ιδιότητες που τα χαρακτηρίζουν και τις λειτουργίες που μπορούν να εκτελούν κατηγοριοποιούνται και περιγράφονται με ένα γενικό τρόπο σε μια **κλάση**. Για κάθε διαφορετικό είδος αντικειμένων πρέπει να ορίσουμε μία ξεχωριστή κλάση.
- Μια κλάση περιγράφει με ένα γενικό (αφηρημένο) τρόπο:
 - (1) τις *ιδιότητες* που χαρακτηρίζουν ένα είδος αντικειμένων ή αλλιώς τις τιμές δεδομένων που χαρακτηρίζουν την *κατάσταση* τους. Οι τιμές των ιδιοτήτων αυτών αποθηκεύονται σε ειδικές μεταβλητές που δηλώνονται στην κλάση και ονομάζονται **πεδία**. Οι τιμές αυτές μπορούν να αλλάζουν κατά την εκτέλεση ενός προγράμματος, αλλάζοντας ταυτόχρονα και την κατάσταση του αντικειμένου.
 - (2) τις *λειτουργίες* που καθορίζουν τη *συμπεριφορά* ενός είδους αντικειμένων και προσδιορίζουν, μεταξύ άλλων, τον τρόπο επεξεργασίας των δεδομένων που περικλείει το κάθε αντικείμενο. Η περιγραφή του τρόπου εκτέλεσης μιας συγκεκριμένης λειτουργίας γίνεται γράφοντας μια ομάδα εντολών στην οποία δίνουμε ένα συγκεκριμένο όνομα και μπορούμε να εκτελούμε για κάθε αντικείμενο της κλάσης, όποτε χρειάζεται. Η υλοποίηση κάθε τέτοιας λειτουργίας ονομάζεται **μέθοδος**. Συχνά επίσης αναφέρουμε ότι στέλνουμε σε ένα αντικείμενο ένα **μήνυμα** και αυτό ανταποκρίνεται εκτελώντας την αντίστοιχη μέθοδο.
- Από τη στιγμή που θα ορίσουμε μια κλάση *μπορούμε να δημιουργήσουμε όσα αντικείμενα, ή αλλιώς στιγμιότυπα, χρειαζόμαστε* για την επίλυση ενός δεδομένου

προβλήματος. Η κλάση αποτελεί το καλούπι βάσει του οποίου κατασκευάζονται τα αντικείμενα. Κάθε στιγμιότυπο/αντικείμενο μιας κλάσης έχει τις ιδιότητες (πεδία) και μπορεί να εκτελέσει τις λειτουργίες (μεθόδους) της κλάσης από την οποία προέρχεται. Όλα τα στιγμιότυπα μιας κλάσης έχουν τα ίδια πεδία, αλλά οι τιμές των πεδίων αυτών και συνεπώς η κατάσταση των αντικειμένων μιας δεδομένης κλάσης μπορεί να διαφέρει.

- Όταν δημιουργούμε ένα νέο αντικείμενο του δίνουμε ένα όνομα προκειμένου να μπορούμε στη συνέχεια να αναφερθούμε σε αυτό μέσω του προγράμματός μας και να του στείλουμε μηνύματα, και επίσης *αρχικοποιούμε* τα πεδία του καθορίζοντας έτσι την αρχική του κατάσταση. Η δημιουργία ενός νέου αντικειμένου γίνεται καλώντας μια ειδική μέθοδο που ορίζεται στην κλάση του αντικειμένου και ονομάζεται **κατασκευαστής**.

Βασική έννοια στον αντικειμενοστρεφή προγραμματισμό αποτελεί και η **κληρονομικότητα**, ο ορισμός δηλαδή μιας κλάσης ως επέκταση κάποιας άλλης. Σε αυτή την περίπτωση η νέα κλάση ονομάζεται **υποκλάση** και εκείνη από την οποία κληρονομεί **υπερκλάση**. Μια υποκλάση κληρονομεί από την υπερκλάση τις ιδιότητες (πεδία) και τις λειτουργίες (μεθόδους) της, και την επεκτείνει με επιπλέον λειτουργίες ή/και πεδία. Η έννοια της κληρονομικότητας μπορεί εύκολα να διδαχθεί σε ειδικά σχεδιασμένα περιβάλλοντα, όπως για παράδειγμα σε ένα προγραμματιστικό μικρόκοσμο, αλλά η διδασκαλία σε μια συμβατική αντικειμενοστρεφή γλώσσα είναι σημαντικά πιο δύσκολη.



Δραστηριότητα 1

Αναπτύξτε μία εισαγωγική εκπαιδευτική δραστηριότητα (ή φύλλο δραστηριοτήτων) που να έχει ως στόχο την ανάδειξη της αντικειμενοστρεφούς τεχνικής σχεδίασης στο πλαίσιο επίλυσης ενός απλού προβλήματος. Η δραστηριότητα αυτή θα έχει ως στόχο τον εντοπισμό των κλάσεων (ή της κλάσης) που απαιτούνται για την αναπαράσταση των αντικειμένων που προϋποθέτει η επίλυση του προβλήματος, καθώς επίσης και των ιδιοτήτων και των λειτουργιών των αντικειμένων που αναπαριστά η κάθε κλάση (και όχι την ανάπτυξη κώδικα).

5. ΔΥΣΚΟΛΙΕΣ, ΠΑΡΑΝΟΗΣΕΙΣ ΚΑΙ ΔΙΔΑΚΤΙΚΕΣ ΠΑΡΕΜΒΑΣΕΙΣ

Η διδασκαλία και εκμάθηση του αντικειμενοστρεφούς προγραμματισμού συνοδεύεται από ποικίλες δυσκολίες και παρανοήσεις. Όταν ο εκπαιδευτικός έχει γνώση των σχετικών

ευρημάτων που έχουν καταγραφεί στη διεθνή βιβλιογραφία είναι σε θέση να σχεδιάσει και να αναπτύξει κατάλληλο εκπαιδευτικό υλικό (παραδείγματα, ερωτήσεις, δραστηριότητες) προκειμένου να αποφευχθεί στο βαθμό που αυτό είναι δυνατό η εμφάνιση συγκεκριμένων δυσκολιών/παρανοήσεων, ή τουλάχιστον να αντιμετωπιστούν οι δυσκολίες αυτές ευκολότερα.

Όσον αφορά στις παρανοήσεις, ή αλλιώς λανθασμένες αντιλήψεις, που έχουν καταγραφεί, ιδιαίτερη βαρύτητα πρέπει να δοθεί στις ακόλουθες:

- *Ταύτιση κλάσης/αντικειμένου*: αρκετοί σπουδαστές συγχέουν τις έννοιες της κλάσης και του αντικειμένου και δεν μπορούν να τις διαχωρίσουν (Holland et al., 1997).

Η σχεδίαση της λύσης σε ένα πρόβλημα με την αντικειμενοστρεφή τεχνική προγραμματισμού προϋποθέτει:

- (1) τον εντοπισμό των οντοτήτων (βασικών στοιχείων) του προβλήματος,
- (2) την μοντελοποίηση κάθε οντότητας, τον εντοπισμό δηλαδή των ιδιοτήτων (πεδίων) και των λειτουργιών (μεθόδων) της με απώτερο στόχο τον ορισμό μιας κλάσης που την αναπαριστά
- (3) τη δημιουργία των απαραίτητων αντικειμένων από κάθε κλάση και την αποστολή των κατάλληλων αντικειμένων (εκτέλεση των κατάλληλων μεθόδων) για την επίλυση του προβλήματος.

Αν ο μαθητής δεν κατανοήσει ότι μια κλάση περιγράφει με αφηρημένο τρόπο ένα είδος αντικειμένων, καθώς επίσης και το γεγονός ότι μπορούμε στο πλαίσιο επίλυσης ενός προβλήματος να δημιουργήσουμε οποιοδήποτε αριθμό αντικειμένων (στιγμιότυπων) από μια κλάση τότε η δυσκολία σχεδίασης μιας αντικειμενοστρεφούς λύσης σε ένα πρόβλημα είναι πολύ μεγάλη.

Ο εντοπισμός της συγκεκριμένης παρανόησης από τον εκπαιδευτικό μπορεί να γίνει εύκολα απευθύνοντας σχετικές ερωτήσεις ανοικτού ή κλειστού τύπου στους μαθητές ή δίνοντας τον προσδιορισμό ενός απλού προβλήματος και ζητώντας από τους μαθητές να καθορίσουν τις απαιτούμενες κλάσεις και αντικείμενα. Επίσης, ο ορισμός από τους μαθητές ενός αντιγράφου (ή περισσότερων αντιγράφων) μιας κλάσης με διαφορετικό όνομα και η δημιουργία ενός αντικειμένου από κάθε μία κλάση για την επίλυση ενός προβλήματος, αντί για τον ορισμό μιας κλάσης και την δημιουργία του κατάλληλου αριθμού αντικειμένων από αυτή, είναι μια συχνή απόρροια της παρανόησης ότι κλάση και αντικείμενο είναι έννοιες ταυτόσημες.

Για την αποφυγή της παρανόησης ότι οι έννοιες κλάση και αντικείμενο είναι έννοιες ταυτόσημες, ο εκπαιδευτικός πρέπει να χρησιμοποιήσει από τα πρώτα μαθήματα παραδείγματα και ασκήσεις, στις οποίες χρησιμοποιείται περισσότερα ένα αντικείμενα από κάθε κλάση.

- *Ταύτιση αντικειμένου/μεταβλητής:* κάθε αντικείμενο αποτελεί απλά ένα «περιτύλιγμα» μιας μεταβλητής.

Σε αρκετές περιπτώσεις οι εκπαιδευτικοί για λόγους απλότητας παρουσιάζουν στα αρχικά μαθήματα παραδείγματα στα οποία χρησιμοποιείται ένα μόνο πεδίο, ή αλλιώς μία μεταβλητή στιγμιοτύπου (instance variable), σε κάθε κλάση. Η χρήση τέτοιου είδους παραδειγμάτων είναι πολύ πιθανό να δημιουργήσει την παρανόηση ότι ένα αντικείμενο είναι απλά ένα «περιτύλιγμα» μιας μεταβλητής χωρίς κάποια λειτουργικότητα, και γι' αυτό θα πρέπει να αποφεύγεται.

- *Τα αντικείμενα είναι απλές εγγραφές χωρίς συμπεριφορά:* δεν γίνεται αντιληπτό ότι η συμπεριφορά ενός αντικειμένου μπορεί να αλλάξει ουσιαστικά ανάλογα με την κατάστασή του (Holland et al., 1997).

Αρκετές φορές οι εκπαιδευτικοί σχεδιάζουν παραδείγματα και ασκήσεις στις οποίες τα αντικείμενα συμπεριφέρονται ως απλές εγγραφές δεδομένων χωρίς καμία απολύτως λειτουργικότητα (συμπεριφορά). Για παράδειγμα, χρησιμοποιείται μια κλάση VideoGame που αναπαριστά ηλεκτρονικά παιχνίδια και έχει ως πεδία τον τίτλο του παιχνιδιού, τον δημιουργό, την πλατφόρμα κτλ. Παρόλο που το θέμα αυτό προσελκύει (ενδεχομένως) το ενδιαφέρον των μαθητών, δεν αναδεικνύει το γεγονός ότι τα αντικείμενα έχουν συμπεριφορά, η οποία μάλιστα μπορεί να αλλάξει ανάλογα με την κατάστασή τους. Τα παραδείγματα που χρησιμοποιούνται και οι ασκήσεις που ανατίθενται στους μαθητές πρέπει να αναφέρονται σε αντικείμενα που περικλείουν δεδομένα, αλλά έχουν και συμπεριφορά που μπορεί μάλιστα να αλλάξει ουσιαστικά ανάλογα με την κατάστασή τους (τις τρέχουσες τιμές δεδομένων των πεδίων τους). Για παράδειγμα, μπορεί να χρησιμοποιηθεί μια κλάση που αναπαριστά ένα Λογαριασμό (ή καλύτερα τον Κουμπάρα) ενός μαθητή. Ένα αντικείμενο της κλάσης αυτής θα συμπεριφερθεί διαφορετικά σε μια αίτηση ανάληψης, ανάλογα με τον αν υπάρχει υπόλοιπο ή όχι στο λογαριασμό.

- *Δεν αξιοποιείται η λειτουργικότητα των μεθόδων* και επαναλαμβάνεται ένα τμήμα κώδικα αντί να ορίζεται μια μέθοδος (Fleury, 2001).

Οι μαθητές συχνά επαναλαμβάνουν πανομοιότυπα τμήματα κώδικα σε διάφορα σημεία ενός προγράμματος, αντί να ορίζουν μια μέθοδο για την υλοποίηση της συγκεκριμένης λειτουργίας και στη συνέχεια να την καλούν στα απαραίτητα σημεία. Ειδικά, όταν το συγκεκριμένο τμήμα κώδικα εκτελείται μέσω άλλων μεθόδων, υπάρχουν μαθητές που ακόμα και αν αντιληφθούν ότι το συγκεκριμένο τμήμα κώδικα θα μπορούσε να αποτελέσει μια μέθοδο προτιμούν αντί αυτού να επαναλαμβάνουν το ίδιο τμήμα κώδικα. Όπως έχουν αναφέρει και άλλοι ερευνητές η κλήση μεθόδων μέσα από άλλες μεθόδους προκαλεί δυσκολίες σε αρκετούς μαθητές.

Ο εκπαιδευτικός θα πρέπει να ετοιμάσει από τα πρώτα μαθήματα κατάλληλες δραστηριότητες προκειμένου να κατανοήσουν οι μαθητές και να αξιοποιήσουν τη

λειτουργικότητα των μεθόδων. Ενδεικτικά, θα μπορούσε να δοθεί στους μαθητές ο ορισμός μιας κλάσης στην οποία επαναλαμβάνεται το ίδιο τμήμα κώδικα και να τεθούν σχετικές ερωτήσεις στους μαθητές, να ακολουθήσει συζήτηση, να εντοπιστεί το πρόβλημα και να βελτιωθεί ο ορισμός της κλάσης με την αντικατάσταση των ίδιων τμημάτων κώδικα από μια κλήση σε μια νέα μέθοδο που θα προστεθεί στην κλάση.

Οι προαναφερθείσες παρανοήσεις είναι πολύ σημαντικές και πρέπει να λαμβάνονται υπόψη για τη σχεδίαση κατάλληλου εκπαιδευτικού υλικού για τα αρχικά μαθήματα εισαγωγής στον αντικειμενοστρεφή προγραμματισμό. Ωστόσο, υπάρχουν πολλές ακόμα δυσκολίες και παρανοήσεις που θα πρέπει να γνωρίζει ο εκπαιδευτικός και να είναι σε θέση να αποτρέψει με τη δημιουργία κατάλληλων διδακτικών καταστάσεων, ή τουλάχιστον να είναι σε θέση να εντοπίσει την ύπαρξή τους εύκολα σχεδιάζοντας κατάλληλα φύλλα ελέγχου. Βέβαια, ορισμένες από τις δυσκολίες και τις παρανοήσεις που συνοψίζονται στη συνέχεια αποφεύγονται όταν για τη διδασκαλία του αντικειμενοστρεφούς προγραμματισμού χρησιμοποιούνται ειδικά περιβάλλοντα. Για παράδειγμα, στους περισσότερους προγραμματιστικούς μικρόκοσμούς η δήλωση των μεταβλητών γίνεται έμμεσα και οι σχετικές δυσκολίες απαλείφονται.

Στη συνέχεια, αναφέρονται δυσκολίες και παρανοήσεις που καταγράφηκαν σε μια μελέτη διάρκειας δύο ετών που πραγματοποιήθηκε με μαθητές (15-16 ετών) σχολείων στο Ισραήλ, χρησιμοποιώντας τη Java και το περιβάλλον BlueJ (Ragonis & Ben-Ari, 2005):

- Μια κλάση είναι μια συλλογή αντικειμένων και όχι ένα πρότυπο δημιουργίας αντικειμένων (παρανόηση).
- Δυσκολία κατανόησης της στατικής φύσης μιας κλάσης και της δυναμικής φύσης ενός αντικειμένου.
- Δυσκολία κατανόησης της διαδικασίας δημιουργίας ενός αντικειμένου.
- Δύο αντικείμενα της ίδιας κλάσης δεν μπορούν να έχουν τις ίδιες τιμές πεδίων (παρανόηση).
- Μπορεί να χρησιμοποιηθεί το ίδιο αναγνωριστικό για δύο αντικείμενα, εφόσον αυτά έχουν κάποια διαφορετική τιμή σε κάποιο από τα πεδία τους (παρανόηση).
- Το αναγνωριστικό ενός αντικειμένου είναι κάποιο από τα πεδία (ιδιότητες) του (παρανόηση).
- Δυσκολία κατανόησης του γεγονότος ότι μια μέθοδος μπορεί να κληθεί για οποιοδήποτε αντικείμενο μιας κλάσης.
- Μπορούμε να καλέσουμε μια μέθοδο για ένα αντικείμενο μόνο μία φορά (παρανόηση).
- Μπορούμε να ορίσουμε μια μέθοδο που προσθέτει μια ιδιότητα/πεδίο σε μια κλάση.

- Δυσκολία κατανόησης της κατηγοριοποίησης των μεθόδων σε κατασκευαστές, μεθόδους πρόσβασης και μετάλλαξης.
- Η κλήση του κατασκευαστή μιας κλάσης δεν είναι απαραίτητη, αφού ο ορισμός του είναι αρκετός για τη δημιουργία ενός αντικειμένου (παρανόηση).
- Οι κατασκευαστές μπορούν να περιλαμβάνουν μόνο εντολές ανάθεσης για την αρχικοποίηση των πεδίων (παρανόηση).
- Η δημιουργία ενός στιγμιότυπου (αντικειμένου) μιας κλάσης αφορά μόνο στην εκτέλεση του κατασκευαστή και όχι στην κατανομή μνήμης (παρανόηση).
- Δυσκολία κατανόησης ενός κενού κατασκευαστή.
- Η αρχικοποίηση ενός πεδίου με μια σταθερά κατά τη δήλωσή του προκαλεί δυσκολία στο διαχωρισμό μεταξύ της κλάσης και του αντικειμένου (παρανόηση).
- Η αρχικοποίηση ενός πεδίου με μια σταθερά σε ένα κατασκευαστή προκαλεί δυσκολία στο διαχωρισμό μεταξύ της κλάσης και του αντικειμένου (παρανόηση).
- Αν τα πεδία αρχικοποιούνται στη δήλωση της κλάσης τότε δεν υπάρχει ανάγκη να δημιουργηθούν αντικείμενα (παρανόηση).
- Μέθοδοι με την ίδια υπογραφή σε διαφορετικές κλάσεις δεν μπορούν να διαχωριστούν (παρανόηση).
- Οι μέθοδοι εκτελούνται σύμφωνα με τη σειρά που εμφανίζονται στον ορισμό της κλάσης (παρανόηση).
- Κάθε μέθοδος μπορεί να κληθεί μόνο μία φορά (παρανόηση).
- Δυσκολία κατανόησης της επίδρασης που έχει η εκτέλεση μιας μεθόδου στην κατάσταση ενός αντικειμένου.
- Δυσκολία κατανόησης της κλήσης μιας μεθόδου από άλλη μέθοδο.
- Δυσκολία κατανόησης της προέλευσης των τιμών των παραμέτρων σε μια μέθοδο.
- Δυσκολία κατανόησης σχετικά με το πού καταλήγει η επιστρεφόμενη τιμή μιας μεθόδου
- Τα αντικείμενα δημιουργούνται από μόνα τους (παρανόηση).
- Το σύστημα δεν επιτρέπει μη λογικές λειτουργίες (παρανόηση).
- Δεν γίνεται αντιληπτό πώς γνωρίζει ο υπολογιστής ποια είναι τα πεδία και οι μέθοδοι μιας κλάσης.
- Δεν γίνεται αντιληπτό πώς η μια κλάση αναγνωρίζει την άλλη.



Δραστηριότητα 2

Αφού μελετήσετε τις συνήθεις δυσκολίες και παρανοήσεις που συνοδεύουν την εκμάθηση του αντικειμενοστρεφούς προγραμματισμού, εξετάστε ξανά το εκπαιδευτικό υλικό που αναπτύξατε στο πλαίσιο της

δραστηριότητας 1 και κάντε τις απαραίτητες αλλαγές, έτσι ώστε:

- να δοθεί έμφαση στις βασικές αντικειμενοστρεφείς έννοιες
- να αντιμετωπιστούν στο μέγιστο δυνατό βαθμό οι σχετικές δυσκολίες των μαθητών
- να αποφευχθεί η δημιουργία των παρανοήσεων που έχουν καταγραφεί στη διεθνή βιβλιογραφία.



Δραστηριότητα 3

Αναπτύξτε ένα φύλλο ελέγχου που να έχει ως στόχο τον εντοπισμό παρανοήσεων (λανθασμένων αντιλήψεων) των μαθητών για τις βασικές έννοιες του αντικειμενοστρεφούς προγραμματισμού: αντικείμενο, κλάση,

μέθοδος, μήνυμα, πεδία. Οι δραστηριότητες που θα αναπτύξετε μπορεί να περιλαμβάνουν, για παράδειγμα, ένα σταυρόλεξο, ένα quiz με ερωτήσεις κλειστού τύπου, ή ακόμα και τη σχεδίαση από τους μαθητές ενός εννοιολογικού χάρτη.

Στη συνέχεια, παρουσιάζονται κάποιες ενδεικτικές ερωτήσεις κλειστού τύπου που μπορείτε να συμπεριλάβετε:

ΕΡΩΤΗΣΗ 1 - Επιλέξτε τη μοναδική σωστή απάντηση:

- (1) Μια κλάση είναι μια συλλογή αντικειμένων.
- (2) Μια κλάση είναι ένα πρότυπο δημιουργίας αντικειμένων.
- (3) Οι έννοιες κλάση και αντικείμενο είναι έννοιες ταυτόσημες.
- (4) όλα τα παραπάνω.

ΕΡΩΤΗΣΗ 2 - Επιλέξτε τη μοναδική λάθος απάντηση:

- (1) Δύο αντικείμενα της ίδιας κλάσης μπορούν να έχουν τις ίδιες τιμές πεδίων.
- (2) Μπορεί να χρησιμοποιηθεί το ίδιο όνομα (αναγνωριστικό) για δύο αντικείμενα, εφόσον αυτά έχουν κάποια διαφορετική τιμή σε κάποιο από τα πεδία τους.
- (3) Από μια κλάση μπορούμε να δημιουργήσουμε όσα αντικείμενα θέλουμε.

ΕΡΩΤΗΣΗ 3 - Επιλέξτε τη σωστή απάντηση:

- (1) Μια μέθοδος μπορεί να κληθεί για οποιοδήποτε αντικείμενο μιας κλάσης.
- (2) Μπορούμε να καλέσουμε μια μέθοδο για ένα αντικείμενο μόνο μία φορά.
- (3) Όλα τα παραπάνω.

ΕΡΩΤΗΣΗ 4 - Επιλέξτε τη λάθος απάντηση:

- (1) Οι μέθοδοι εκτελούνται σύμφωνα με τη σειρά που εμφανίζονται στον ορισμό της κλάσης.
- (2) Κάθε μέθοδος μπορεί να κληθεί μόνο μία φορά.
- (3) Μέσω μιας μεθόδου μπορεί να γίνει κλήση άλλης μεθόδου.
- (4) Τα δύο πρώτα.
- (5) Τα τρία πρώτα.

ΕΡΩΤΗΣΗ 5 - Επιλέξτε τη μοναδική σωστή απάντηση:

- (1) Η εκτέλεση μιας μεθόδου μπορεί να αλλάξει την κατάσταση ενός αντικειμένου.
- (2) Το αποτέλεσμα της εκτέλεσης μιας συγκεκριμένης μεθόδου είναι πάντα το ίδιο ανεξάρτητα από την κατάσταση του αντικειμένου.
- (3) Όλα τα παραπάνω.

ΕΡΩΤΗΣΗ 6 - Επιλέξτε τη μοναδική σωστή απάντηση:

- (1) Ένα αντικείμενο ανταποκρίνεται σε κάποιο μήνυμα αρκεί αυτό να έχει οριστεί σε κάποια από τις κλάσεις που υπάρχουν στο πρόγραμμα.
- (2) Ένα αντικείμενο ανταποκρίνεται σε ένα μήνυμα μόνο εφόσον έχει οριστεί η αντίστοιχη μέθοδος στην κλάση από την οποία προέρχεται το αντικείμενο.
- (3) Ένα αντικείμενο ανταποκρίνεται σε ένα μήνυμα πάντα με τον ίδιο ακριβώς τρόπο.
- (4) Όλα τα παραπάνω.

6. ΔΙΔΑΚΤΙΚΑ ΠΡΟΒΛΗΜΑΤΑ

Στην ενότητα αυτή παρουσιάζονται τα σημαντικότερα γενικά προβλήματα που έχουν καταγραφεί στη διεθνή βιβλιογραφία σχετικά με τη διδασκαλία του αντικειμενοστρεφούς προγραμματισμού. Ο εκπαιδευτικός θα πρέπει να γνωρίζει τα προβλήματα αυτά προκειμένου να σχεδιάσει ένα αποτελεσματικό μάθημα εισαγωγής στη συγκεκριμένη

τεχνική προγραμματισμού.

Μετάβαση από τον διαδικαστικό στον αντικειμενοστρεφή προγραμματισμό

Αρκετές εργασίες έχουν εκπονηθεί με αντικείμενο τη σειρά με την οποία είναι προτιμότερο να εισάγονται οι σπουδαστές στις διάφορες τεχνικές προγραμματισμού, και ειδικότερα τη σειρά με την οποία πρέπει να διδάσκονται την τεχνική του αντικειμενοστρεφούς και του διαδικαστικού/δομημένου προγραμματισμού. Αν και οι απόψεις δίστανται, η πλειονότητα των διδασκόντων/ερευνητών υποστηρίζει ότι:

- η αντικειμενοστρεφής προσέγγιση προγραμματισμού πρέπει να παρουσιάζεται στα αρχικά στάδια διδασκαλίας του προγραμματισμού (Alphonse & Ventura, 2002; Bruce et al., 2001; Christensen & Caspersen, 2002; Kolling & Rosenberg, 2001)
- οι σπουδαστές αντιμετωπίζουν περισσότερες δυσκολίες όταν μεταβαίνουν από τον κατηγορηματικό (imperative) – διαδικαστικό (procedural) προγραμματισμό στον αντικειμενοστρεφή, ενώ το αντίστροφο δεν ισχύει (Decker & Hirshfield, 1994; Hadjerrouit, 1998; Tempete, 1991; Wick, 1995).

Όταν οι μαθητές έχουν προηγούμενη εμπειρία σε κάποια διαδικαστική γλώσσα (ή ψευδογλώσσα) η εκμάθηση των αρχών του αντικειμενοστρεφούς προγραμματισμού καθίσταται δυσκολότερη. Οι σημαντικότερες δυσκολίες που παρουσιάζονται είναι οι εξής:

- οι μαθητές *δυσκολεύονται να αλλάξουν τρόπο σκέψης* και να αρχίσουν να σχεδιάζουν αντικειμενοστρεφείς λύσεις σε προβλήματα. Παρά τους ισχυρισμούς ότι η αντικειμενοστρεφής προσέγγιση επίλυσης προβλημάτων είναι πιο εύκολη -φυσική για την ακρίβεια-, ο νέος τρόπος σκέψης που απαιτεί δεν γίνεται εύκολα κατανοητός από σπουδαστές που έχουν εμπειρία στην επίλυση προβλημάτων με μια διαδικαστική γλώσσα προγραμματισμού (Tempete, 1991).
- *δεν αξιοποιούνται τα πλεονεκτήματα του αντικειμενοστρεφούς προγραμματισμού.* Σπουδαστές με προηγούμενη εμπειρία σε διαδικαστικές γλώσσες προγραμματισμού, κατά την εισαγωγή τους στη Java αντιμετώπισαν δυσκολίες με τις έννοιες του αντικειμενοστρεφούς προγραμματισμού και συγκεκριμένα με την αξιοποίησή τους. Για παράδειγμα, οι σπουδαστές παρουσίασαν την τάση να χρησιμοποιούν τις μεθόδους ως διαδικασίες, αγνοώντας το ρόλο τους στα πλαίσια του αντικειμενοστρεφούς προγραμματισμού (Hadjerrouit, 1998; Hadjerrouit, 1999).

Στην περίπτωση λοιπόν που οι μαθητές έχουν προηγούμενη εμπειρία σε κάποια διαδικαστική (ψευδο)γλώσσα είναι πολύ σημαντικό ο εκπαιδευτικός να αφιερώσει χρόνο

στο πρώτο μάθημα για να αντιπαραθέσει τα βασικά χαρακτηριστικά των τεχνικών του διαδικαστικού και του αντικειμενοστρεφούς προγραμματισμού. Είναι απαραίτητο να κατανοήσουν οι μαθητές ότι:

- για την επίλυση ενός προβλήματος με την τεχνική του διαδικαστικού προγραμματισμού εντοπίζουμε τα υποπροβλήματα από τα οποία αποτελείται το πρόβλημα και αναπτύσσουμε ένα υποπρόγραμμα για κάθε ένα από αυτά.
- για την επίλυση ενός προβλήματος με την τεχνική του αντικειμενοστρεφούς προγραμματισμού εντοπίζουμε τις οντότητες, τα διακριτά στοιχεία ενός προβλήματος και για κάθε ένα από αυτά ορίζουμε μία κλάση. Μία κλάση περιλαμβάνει δεδομένα (αποθηκεύονται σε ειδικές μεταβλητές που ονομάζονται πεδία) και λειτουργίες για την επεξεργασία τους (μέθοδοι) και αποτελεί ένα πρότυπο με βάση το οποίο μπορούμε να δημιουργήσουμε οποιοδήποτε αριθμό αντικειμένων απαιτείται για την επίλυση ενός προβλήματος. Στα αντικείμενα στέλνουμε μηνύματα μέσω του προγράμματός μας για να εκτελέσουν διάφορες λειτουργίες (μεθόδους). Ένα αντικείμενο έχει συγκεκριμένη κατάσταση, η οποία αλλάζει με την εκτέλεση των διαφόρων μεθόδων της κλάσης στην οποία ανήκει και συμπεριφορά που καθορίζεται από τις μεθόδους που μπορεί να εκτελέσει.
- τα υποπρογράμματα και οι μέθοδοι δεν είναι έννοιες ταυτόσημες. Ένα αντικείμενο μπορεί να εκτελεί μόνο τις μεθόδους που έχουν οριστεί στην κλάση από την οποία προέρχεται.

Η αντικειμενοστραφής τεχνική ανάπτυξης προγραμμάτων είναι δύσκολη

Η αντικειμενοστρεφής τεχνική ανάπτυξης προγραμμάτων θεωρείται πιο αφηρημένη σε σχέση με την τεχνική του δομημένου προγραμματισμού, απαιτεί νέους τρόπους σκέψης και είναι πιο απαιτητική όσον αφορά τις διαδικασίες της ανάλυσης και σχεδίασης.

Επίσης, από μελέτη που διεξήχθη προέκυψε ότι τα νοητά μοντέλα που δημιουργούν οι φοιτητές για τα αντικειμενοστρεφή και τα κατηγορηματικά προγράμματα διαφέρουν ουσιαστικά (Ramalingam & Wiedenbeck, 1997). Συγκεκριμένα, οι φοιτητές κατανόησαν σε μεγαλύτερο βαθμό τα κατηγορηματικά προγράμματα. Τα νοητά μοντέλα που δημιούργησαν οι φοιτητές για τα κατηγορηματικά προγράμματα επικεντρώθηκαν στις στοιχειώδεις διαδικασίες (elementary operations) και τη ροή ελέγχου (control flow) του προγράμματος που περιγράφονται από τον πηγαίο κώδικα, δημιούργησαν δηλαδή ένα μοντέλο για το

πρόγραμμα (program model). Αντίθετα, τα νοητά μοντέλα που δημιούργησαν οι φοιτητές για τα αντικειμενοστρεφή προγράμματα επικεντρώθηκαν περισσότερο στη ροή των δεδομένων (data flow) και τη λειτουργία του προγράμματος (function knowledge), δημιούργησαν δηλαδή ένα μοντέλο για τη λειτουργία του προγράμματος συνολικά (domain model).

Είναι λοιπόν σημαντικό ο εκπαιδευτικός να χρησιμοποιήσει τα κατάλληλα εργαλεία και να σχεδιάσει εκπαιδευτικές δραστηριότητες προκειμένου οι μαθητές να κατανοήσουν τόσο τα επιμέρους συστατικά στοιχεία ενός αντικειμενοστρεφούς προγράμματος όσο και τη λειτουργία του συνολικά. Αυτό μπορεί να επιτευχθεί χρησιμοποιώντας περιβάλλοντα που υποστηρίζουν τη βηματική εκτέλεση των προγραμμάτων και δυνατότητες οπτικοποίησης των αποτελεσμάτων της εκτέλεσης και φυσικά την εμπλοκή των μαθητών σε δραστηριότητες που απαιτούν τον εντοπισμό, την κατανόηση και την καταγραφή στοιχείων που αφορούν τόσο στον κώδικα ενός προγράμματος και στη δομή του, όσο και στη λειτουργία του συνολικά.



Δραστηριότητα 4

Λαμβάνοντας υπόψη τα προαναφερθέντα διδακτικά προβλήματα, καθώς επίσης και το γεγονός ότι η διδασκαλία του αντικειμενοστρεφούς προγραμματισμού θα πραγματοποιηθεί σε μαθητές με προηγούμενη εμπειρία στον διαδικαστικό προγραμματισμό, να σχεδιάσετε εισαγωγικές δραστηριότητες με στόχο:

- να κατανοήσουν οι μαθητές τις βασικές διαφορές σχεδίασης της λύσης σε ένα πρόβλημα με τις τεχνικές του διαδικαστικού και του αντικειμενοστρεφούς προγραμματισμού
- να μην ταυτίσουν την έννοια της διαδικασίας/συνάρτησης ενός διαδικαστικού προγράμματος με την έννοια της μεθόδου σε ένα αντικειμενοστρεφές πρόγραμμα.

7. ΧΡΗΣΗ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΛΟΓΙΣΜΙΚΟΥ

Γενικές δυσκολίες εκμάθησης του προγραμματισμού

Η διδασκαλία του προγραμματισμού, ανεξάρτητα από την τεχνική προγραμματισμού που χρησιμοποιείται, συνοδεύεται από αρκετές δυσκολίες (Ξυνογάλος κ.α., 2000):

- (1) Οι γλώσσες προγραμματισμού γενικού σκοπού διαθέτουν κατά κανόνα ένα μεγάλο ρεπερτόριο εντολών και είναι πολύπλοκες.
- (2) Η προσοχή των μαθητών επικεντρώνεται στην εκμάθηση της σύνταξης της γλώσσας και όχι στην ανάπτυξη ικανοτήτων επίλυσης προβλημάτων.

- (3) Δεν υπάρχει, κατά κανόνα, επαρκής στήριξη του μαθητή στην κατανόηση των βασικών ενεργειών και δομών ελέγχου, αφού το περιβάλλον προγραμματισμού συνήθως δεν παρέχει δυνατότητες οπτικοποίησης (η διαδικασία εκτέλεσης ενός προγράμματος δεν είναι ορατή από τον σπουδαστή), ή ο τρόπος χρήσης τους δεν είναι προφανής στους μαθητές.
- (4) Οι εμπορικοί μεταγλωττιστές δεν ικανοποιούν τις ανάγκες των αρχάριων προγραμματιστών, με σημαντικότερο μειονέκτημα την αναφορά δυσνόητων μηνυμάτων λάθους.
- (5) Η διανοητική πολυπλοκότητα που απαιτεί η εκφορά ενός αλγορίθμου σε μια γλώσσα προγραμματισμού είναι μεγάλη, λόγω της «φύσης» της γλώσσας.
- (6) Η επίλυση ενδιαφερόντων προβλημάτων απαιτεί την εκμάθηση ενός μεγάλου υποσυνόλου της γλώσσας και την ανάπτυξη αρκετά μεγάλων προγραμμάτων, απαιτεί δηλαδή τη επικέντρωση της προσοχής στην εκμάθηση της γλώσσας (βλέπε παραπάνω σημείο 2).

Είναι προφανές ότι η διδασκαλία του προγραμματισμού σε μαθητές πρέπει να γίνεται χρησιμοποιώντας ειδικές γλώσσες και εκπαιδευτικά περιβάλλοντα προγραμματισμού. Για τη διδασκαλία του δομημένου προγραμματισμού σε μαθητές της Β/θμιας Εκπαίδευσης έχουν αναπτυχθεί στη χώρα μας αρκετά εργαλεία που επιτρέπουν την εισαγωγή σε βασικές έννοιες χρησιμοποιώντας είτε τη γνωστή ψευδογλώσσα «ΓΛΩΣΣΑ» (Γλωσσομάθεια, Διερμηνευτής της Γλώσσας) είτε διάγραμμα ροής (Δημιουργός Διαγραμμάτων Ροής, Διάγραμμα Ροής) και χρησιμοποιούν την ελληνική γλώσσα για την επικοινωνία συστήματος-χρήστη. Για τη διδασκαλία του αντικειμενοστρεφούς προγραμματισμού, ωστόσο, οι επιλογές είναι πολύ πιο περιορισμένες.

Στην παρούσα ενότητα παρουσιάζεται μια διδακτική προσέγγιση για τον αντικειμενοστρεφή προγραμματισμό, η οποία βασίζεται στους προγραμματιστικούς μικρόκοσμους. Αρχικά, παρουσιάζονται τα ιδιαίτερα χαρακτηριστικά των προγραμματιστικών μικρόκοσμων που τους καθιστούν ιδανικούς για την εισαγωγή στον προγραμματισμό και ειδικότερα τον αντικειμενοστρεφή προγραμματισμό. Στη συνέχεια, παρουσιάζονται κάποια βασικά χαρακτηριστικά των προγραμματιστικών μικρόκοσμων που υπάρχουν διαθέσιμοι για τη διδασκαλία του αντικειμενοστρεφούς προγραμματισμού. Ιδιαίτερη έμφαση δίνεται στον προγραμματιστικό μικρόκοσμο `objectKarel`, ο οποίος ενσωματώνει στο περιβάλλον (εκτός από τα γνωστά χαρακτηριστικά των μικρόκοσμων) την απαραίτητη θεωρία για τη διδασκαλία των βασικών εννοιών του αντικειμενοστρεφούς προγραμματισμού και των βασικών αλγοριθμικών δομών (ακολουθίας, επιλογής, επανάληψης), καθώς επίσης και μια

σειρά δραστηριοτήτων. Επιπλέον, στον μικρόκοσμο objectKarel υπάρχει η δυνατότητα διαλόγου με το σύστημα στην ελληνική γλώσσα.

Τα πλεονεκτήματα των προγραμματιστικών μικρόκοσμων

Η βασική ιδέα των μικρόκοσμων (microworlds) και των μικρογλωσσών (mini-languages) προγραμματισμού είναι η δημιουργία μιας μικρής και απλής γλώσσας προγραμματισμού για τη στήριξη των πρώτων βημάτων της εκμάθησης του προγραμματισμού. Η πλειοψηφία των μικρόκοσμων προγραμματισμού ενσωματώνει ένα ανοιχτό περιβάλλον που βασίζεται σε κάποιο φυσικό μοντέλο, ενώ ο χρήστης ελέγχει ένα πρωταγωνιστή που «ζει» στο περιβάλλον αυτό. Ο πρωταγωνιστής μπορεί να είναι μια χελώνα, ένα ρομπότ ή κάποια άλλη οντότητα. Στην περίπτωση του αντικειμενοστρεφούς προγραμματισμού, ένας μικρόκοσμος ενσωματώνει ένα ή περισσότερα μοντέλα (κλάσεις) πρωταγωνιστών, τα οποία χρησιμοποιούνται για τη δημιουργία του κατάλληλου αριθμού πρωταγωνιστών για ένα πρόβλημα. Ο μαθητής μαθαίνει βασικές έννοιες του αντικειμενοστρεφούς προγραμματισμού δημιουργώντας τον κατάλληλο αριθμό πρωταγωνιστών και στέλνοντας σε αυτούς τα απαραίτητα μηνύματα για την επίλυση ενός προβλήματος. Οι μέθοδοι είναι ιδιαίτερα απλοποιημένες, ενώ το αποτέλεσμα της εκτέλεσης τους είναι ορατό στην οθόνη. Συγκεκριμένα, ο χρήστης βλέπει τον πρωταγωνιστή ή τους πρωταγωνιστές του μικρόκοσμου να ανταποκρίνονται σε κάθε μήνυμα εκτελώντας την αντίστοιχη μέθοδο, μεταβάλλοντας κατ' αυτό τον τρόπο την κατάστασή τους και την κατάσταση του περιβάλλοντος στο οποίο ζουν. Αν και το σύνολο των διαθέσιμων μηνυμάτων/μεθόδων είναι μικρό, ωστόσο υπάρχει η δυνατότητα επίλυσης τόσο απλών όσο και αρκετά πολύπλοκων προβλημάτων. Εκτός από τις εντολές ελέγχου τις οποίες εκτελούν οι πρωταγωνιστές του μικρόκοσμου, υπάρχουν και αρκετά ερωτήματα (συναρτήσεις) στα οποία μπορούν να απαντήσουν, έχουν δηλαδή τη δυνατότητα να ελέγχουν την κατάσταση του κόσμου όπου «ζουν». Επίσης, οι περισσότεροι μικρόκοσμοι περιλαμβάνουν όλες τις βασικές δομές ελέγχου και ένα μηχανισμό δημιουργίας νέων κλάσεων, ή τουλάχιστον προσθήκης νέων μεθόδων στις υπάρχουσες κλάσεις.

Οι μικρόκοσμοι μπορούν να χρησιμοποιηθούν αποτελεσματικά στα πλαίσια εκμάθησης εννοιών του αντικειμενοστρεφούς προγραμματισμού και στην απόκτηση δεξιοτήτων επίλυσης προβλημάτων και αλγοριθμικού τρόπου σκέψης. Η διδασκαλία του αντικειμενοστρεφούς προγραμματισμού με την προσέγγιση των μικρόκοσμων παρουσιάζει αρκετά πλεονεκτήματα:

- Η γλώσσα προγραμματισμού αποτελείται από ένα περιορισμένο ρεπερτόριο εντολών με απλή σύνταξη και σημασιολογία.

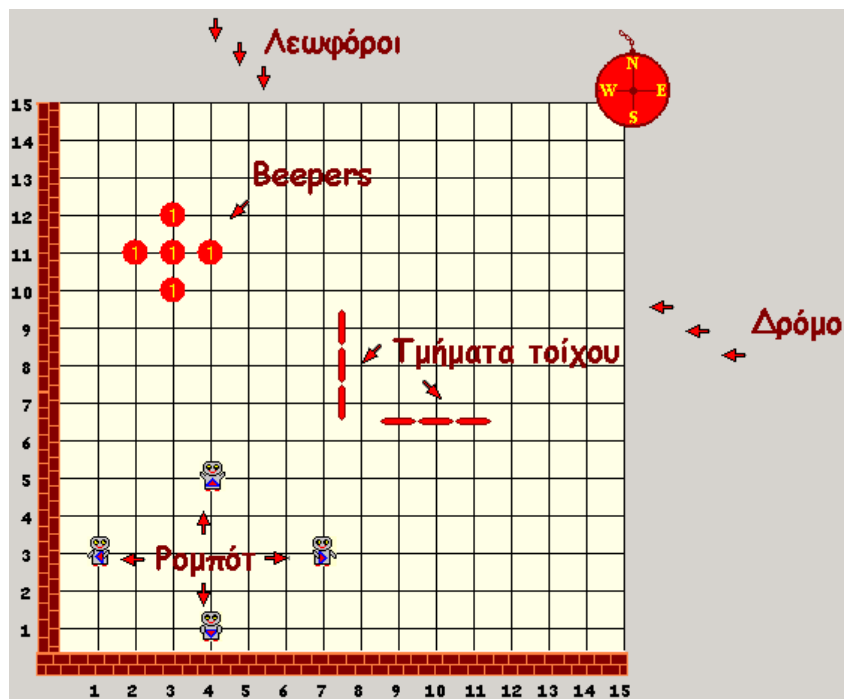
- Βασίζονται σε υπαρκτά μοντέλα που είναι ήδη γνωστά στο σπουδαστή, μειώνοντας έτσι δραματικά τη διανοητική «απόσταση» ανάμεσα στα νοητά μοντέλα ή την περιγραφή σε φυσική γλώσσα των αλγορίθμων και στην περιγραφή τους στη γλώσσα προγραμματισμού.
- Τα προβλήματα που καλούνται να λύσουν οι σπουδαστές παρουσιάζουν ιδιαίτερο ενδιαφέρον.
- Η εκτέλεση ενός προγράμματος είναι ορατή, αποκαλύπτοντας έτσι τη σημασία των διδασκόμενων δομών, καθώς και τις έννοιες που σχετίζονται με τη δομή και την εκτέλεση των προγραμμάτων.
- Υπάρχει δυνατότητα προσαρμογής του μικρόκοσμου στις ανάγκες του κοινού στο οποίο απευθύνεται.

Μικρόκοσμοι βασισμένοι στο ρομπότ Karel

Ένας από τους πιο δημοφιλείς μικρόκοσμους προγραμματισμού όπως επισημαίνουν αρκετοί ερευνητές (Brusilovsky, Calabrese, Hvorenky, Miller, Kouchnirenko, 1997) είναι ο “*Karel the Robot*” (Richard E. Pattis, 1995) που σχεδιάστηκε προκειμένου να χρησιμοποιηθεί για το μάθημα της εισαγωγής στον δομημένο προγραμματισμό. Ο Karel, ο πρωταγωνιστής του μικρόκοσμου, εκτελεί διάφορες αποστολές (προγράμματα) σε ένα κόσμο που αποτελείται από οριζόντιους δρόμους και κάθετες λεωφόρους. Στον κόσμο του Karel μπορεί να υπάρχουν τμήματα τοίχου που τοποθετούνται μεταξύ των διασταυρώσεων δημιουργώντας εμπόδια (π.χ. λαβύρινθος) που καλείται να ξεπεράσει ο Karel και beepers, μικροί πλαστικοί κώνοι που παράγουν ένα ήχο (μπιπ). Ο Karel έχει τη δυνατότητα να κινείται προς την τρέχουσα κατεύθυνση κατά 1 μπλοκ, να στρίβει κατά 90 μοίρες, να εντοπίζει beepers που βρίσκονται στην ίδια διασταύρωση μ’ αυτόν, να εντοπίζει τοίχους που βρίσκονται μπροστά του σε απόσταση μισού μπλοκ χρησιμοποιώντας μια κάμερα, να σηκώνει και να κατεβάζει beepers με το μηχανικό του χέρι και τέλος μπορεί να καθορίζει προς ποια κατεύθυνση βλέπει χρησιμοποιώντας την πυξίδα του. Η γλώσσα προγραμματισμού εκτός από τις εντολές ελέγχου του Karel (move, turnleft, pickbeeper, putbeeper) περιλαμβάνει όλες τις βασικές δομές ελέγχου (if, if-else, while, loop).

Η μεταγενέστερη έκδοση του ρομπότ Karel, γνωστή ως Karel++ (Bergin et al., 1997), υποστηρίζει τη διδασκαλία του αντικειμενοστρεφούς προγραμματισμού. Στον Karel++ υπάρχουν δύο μοντέλα (κλάσεις) ρομπότ που ανταποκρίνονται σε ένα περιορισμένο αριθμό μηνυμάτων. Οι μαθητές μπορούν να δημιουργήσουν ρομπότ των δύο βασικών μοντέλων, αλλά για πιο απαιτητικές αποστολές μπορούν να δημιουργήσουν και τα δικά τους μοντέλα

ρομπότ. Τα νέα μοντέλα ρομπότ κληρονομούν όλες τις δυνατότητες των βασικών μοντέλων και τις επεκτείνουν. Στη συνέχεια, περιγράφονται τα βασικά χαρακτηριστικά δύο υλοποιήσεων που διατίθενται ελεύθερα, ενώ στην επόμενη ενότητα περιγράφεται μια τρίτη υλοποίηση που εκτός από τον μικρόκοσμο ενσωματώνει και διδακτικό υλικό.



Εικόνα 1. Ο μικρόκοσμος των ρομπότ

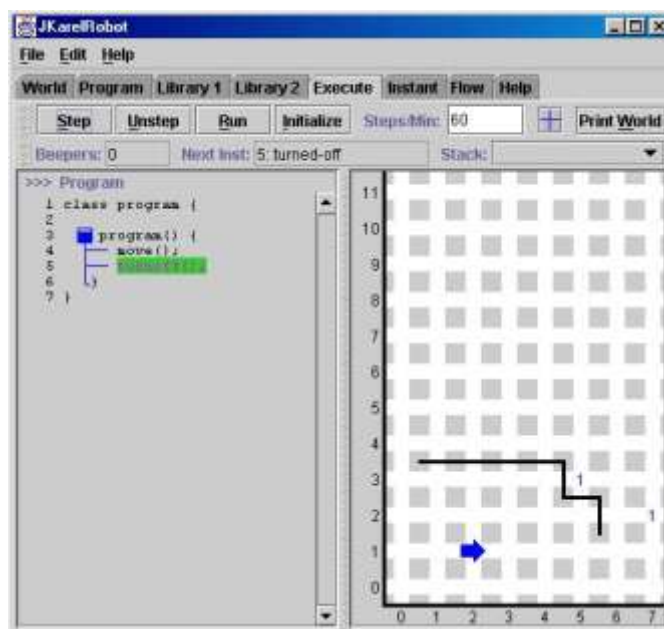
JKarelRobot

<http://math.otterbein.edu/home/Class/Csc120/WebPages/KarelStart.html>

Ο μικρόκοσμος JKarelRobot (Buck and Stucki, 2000) βασίζεται στο ρομπότ Karel (Pattis et al. 1995) και παρέχει τη δυνατότητα προγραμματισμού του ομώνυμου ρομπότ χρησιμοποιώντας μια γλώσσα προγραμματισμού που μοιάζει με την Pascal, τη Java ή τη Lisp. Ο μικρόκοσμος JKarelRobot αποτελείται από έξι διαφορετικές καρτέλες (Εικόνα 2):

- (1) την καρτέλα *World*, στην οποία ο σπουδαστής μπορεί να δημιουργήσει και να ανακαλέσει καταστάσεις του κόσμου, καθώς επίσης και να καθορίσει την αρχική κατάσταση του Karel που είναι το μοναδικό ρομπότ που μπορεί να χρησιμοποιηθεί,
- (2) την καρτέλα *Program* όπου ο σπουδαστής αναπτύσσει, μεταγλωττίζει και διαχειρίζεται τα προγράμματα του,
- (3) την καρτέλα *Execute* στην οποία ο σπουδαστής εκτελεί τα προγράμματά του βήμα προς βήμα είτε προς τα εμπρός είτε προς τα πίσω,

- (4) την καρτέλα *Instant* όπου ο σπουδαστής μπορεί να γράψει μία ή περισσότερες εντολές και να τις εκτελέσει όπως σε ένα ερμηνευτή,
- (5) την καρτέλα *Flow* που χρησιμοποιείται για την ανάπτυξη διαγραμμάτων ροής για υπάρχοντα προγράμματα,
- (6) την καρτέλα *Help* στην οποία παρέχεται ένα πολύ σύντομο tutorial.



Εικόνα 2. Ο μικρόκοσμος JKarelRobot

Στην ουσία, στο περιβάλλον του JKarelRobot η δημιουργία της αρχικής κατάστασης του κόσμου, η ανάπτυξη του προγράμματος και η εκτέλεση του πραγματοποιούνται σε 3 διαφορετικές καρτέλες. Επιπλέον, παρόλο που ο σπουδαστής χρειάζεται σε κάθε καρτέλα να εκτελέσει μια συγκεκριμένη ακολουθία ενεργειών πατώντας κάποια κουμπιά, όλες οι καρτέλες και τα κουμπιά είναι συνεχώς ενεργά και δεν παρέχεται καμία καθοδήγηση στον χρήστη.

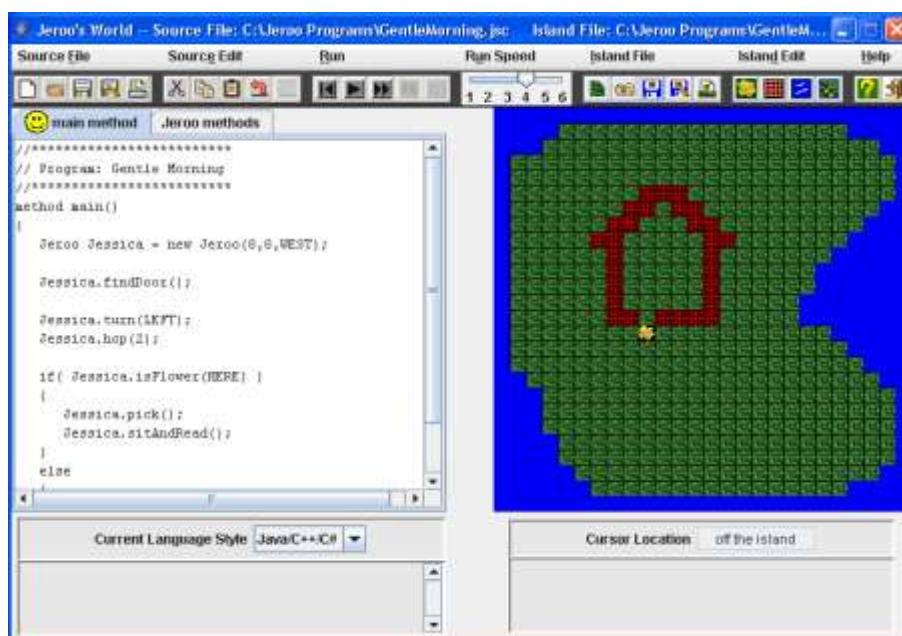
Όπως ήδη αναφέρθηκε στην αρχή, ο σπουδαστής έχει τη δυνατότητα να αναπτύξει τα προγράμματά του σε μια Java-like γλώσσα. Ωστόσο, στα προγράμματα δεν χρησιμοποιούνται εντολές δημιουργίας και αρχικοποίησης αντικειμένων, αφού το περιβάλλον υποστηρίζει τη χρήση ενός μόνο αντικειμένου (ρομπότ) το οποίο αρχικοποιείται στο πλαίσιο καθορισμού της αρχικής κατάστασης του κόσμου. Ο περιορισμός αυτός, δηλαδή η ύπαρξη ενός μόνο αντικειμένου στο περιβάλλον του JKarelRobot μπορεί να οδηγήσει τους σπουδαστές στην παρανόηση ότι κλάση και αντικείμενο είναι έννοιες ταυτόσημες. Επίσης, στο tutorial του JKarelRobot δεν γίνεται καμία αναφορά στις βασικές έννοιες του

αντικειμενοστρεφούς παραδείγματος προγραμματισμού: αντικείμενο, μήνυμα, κλάση, κληρονομικότητα, πολυμορφισμός.

Jeroo

<http://home.cc.gatech.edu/dorn/jeroo>

Ο μικρόκοσμος Jeroo (Sanders and Dorn, 2003) βασίζεται στο ρομπότ Karel, αλλά χρησιμοποιεί μια γλώσσα προγραμματισμού που επιτρέπει μια πιο εύκολη μετάβαση στη C++ και τη Java. Το γραφικό ενδιάμεσο αποτελείται από ένα μόνο παράθυρο (Εικόνα 3), ενώ το περιβάλλον υποστηρίζει την ανάπτυξη αναδρομικών μεθόδων και τη βηματική εκτέλεση των προγραμμάτων.



Εικόνα 3. Ο μικρόκοσμος Jeroo

Ωστόσο, ο μικρόκοσμος Jeroo παρουσιάζει τους εξής περιορισμούς: (1) υπάρχει μία μόνο κλάση και δεν υπάρχει δυνατότητα δημιουργίας νέων κλάσεων, (2) οι σπουδαστές μπορούν να δημιουργήσουν μέχρι τέσσερα αντικείμενα της υπάρχουσας κλάσης, (3) δεν υποστηρίζεται η έννοια της κληρονομικότητας, (4) οι σπουδαστές μπορούν να επεκτείνουν την κλάση Jeroo με void μεθόδους, αλλά δεν μπορούν να αναπτύξουν κατηγορήματα (predicates). Σύμφωνα βέβαια με τους δημιουργούς του Jeroo, οι παραπάνω δυνατότητες δεν ενσωματώθηκαν στον μικρόκοσμο γιατί θεωρήθηκαν περιττές, αφού σκοπός του μικρόκοσμου είναι οι σπουδαστές να μάθουν τις βασικές έννοιες που σχετίζονται με τη χρήση αντικειμένων για την επίλυση προβλημάτων και την ανάπτυξη μεθόδων που καθορίζουν τη συμπεριφορά των αντικειμένων.

objectKarel

<http://users.uom.gr/~stelios/objectKarel.zip>

Ο μικρόκοσμος objectKarel βασίζεται στον Karel++ και υποστηρίζει τη διδασκαλία και εκμάθηση όλων των βασικών αντικειμενοστρεφών εννοιών: αντικείμενο, μήνυμα, κλάση, κληρονομικότητα, πολυμορφισμός και υποσκέλιση. Επιπλέον, υποστηρίζει τις καθιερωμένες δομές επιλογής και επανάληψης. Ο μικρόκοσμος objectKarel ενσωματώνει:

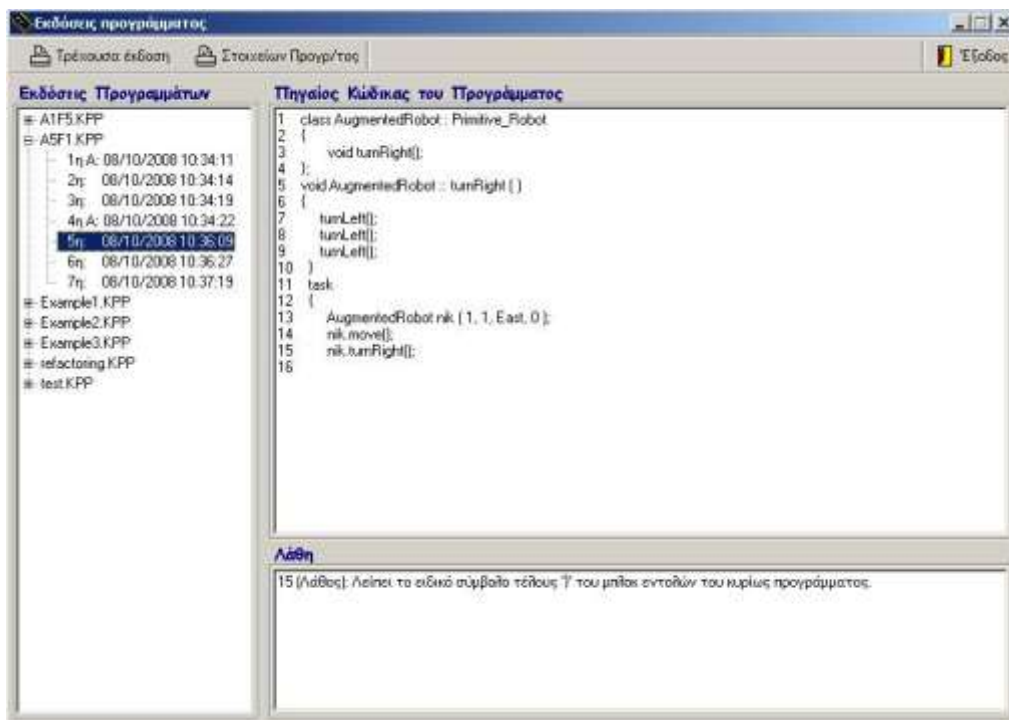
- το απαραίτητο *διδακτικό υλικό* (σύντομη και περιεκτική θεωρία, εγχειρίδιο χρήσης)
- *δραστηριότητες* για την εξοικείωση των μαθητών με τις έννοιες, πριν την εφαρμογή τους στο πλαίσιο ανάπτυξης προγραμμάτων
- ένα συντάκτη δομής για την εύκολη ανάπτυξη προγραμμάτων και την αποφυγή επικέντρωσης στις λεπτομέρειες της γλώσσας προγραμματισμού
- τη δυνατότητα της *επεξηγηματικής οπτικοποίησης*, εμφάνισης δηλαδή μηνυμάτων σε φυσική γλώσσα για την τρέχουσα κάθε φορά εντολή κατά τη βηματική εκτέλεση των προγραμμάτων.

Η διδακτική λογική που υιοθετήθηκε κατά τη σχεδίαση του μικρόκοσμου, αλλά και την προετοιμασία του σχετικού διδακτικού υλικού, αναλύεται διεξοδικά στα σχετικά επιμορφωτικά σενάρια.

Στο σημείο αυτό αξίζει ωστόσο να αναφερθούν κάποιες επιπρόσθετες δυνατότητες που παρέχονται στον εκπαιδευτικό. Για τα περισσότερα προβλήματα που καλούνται να λύσουν οι μαθητές πρέπει να υπάρχει μια συγκεκριμένη αρχική κατάσταση του κόσμου, η οποία δημιουργείται χρησιμοποιώντας *τεχνικές άμεσης διαχείρισης* (κάνοντας κλικ μεταξύ διασταυρώσεων εμφανίζονται τμήματα τοίχου και κάνοντας κλικ πάνω σε διασταυρώσεις beepers). Επίσης, υπάρχει *δυνατότητα επέμβασης στην τρέχουσα κατάσταση του κόσμου ανά πάσα στιγμή*. Ο εκπαιδευτικός μπορεί, στο πλαίσιο παρουσίασης παραδειγμάτων, να αλλάξει την κατάσταση του κόσμου κατά την εκτέλεση ενός προγράμματος και να ζητήσει από τους μαθητές να προσδιορίσουν ποιο θα είναι το αποτέλεσμα της εκτέλεσής του από εκείνο το σημείο και μετά, ελέγχοντας έτσι το βαθμό κατανόησης των διδασκόμενων εννοιών. Επίσης, ο *μαθητής έχει τη δυνατότητα να πειραματιστεί και να διερευνήσει* ποιο θα είναι το αποτέλεσμα της εκτέλεσης ενός τμήματος κώδικα ή ενός προγράμματος για μια διαφορετική κατάσταση του κόσμου και να λύσει τυχόν απορίες, χωρίς να απαιτείται να διακόψει την εκτέλεση του προγράμματος, να αλλάξει την κατάσταση του κόσμου και να εκτελέσει ξανά το πρόγραμμα από την αρχή.

Επίσης, στον μικρόκοσμο objectKarel έχει ενσωματωθεί η δυνατότητα καταγραφής των

ενεργειών των σπουδαστών κατά την ανάπτυξη και αποσφαλμάτωση των προγραμμάτων. Συγκεκριμένα, κάθε φορά που ο μαθητής μεταγλωττίζει ένα πρόγραμμα το σύστημα αποθηκεύει αυτόματα τόσο τον πηγαίο κώδικα όσο και τα αποτελέσματα της μεταγλώττισης. Η ιστορία των μεταγλωττίσεων παρουσιάζεται σε ένα διαφορετικό παράθυρο (Εικόνα 4) με τη μορφή ενός δένδρου δύο επιπέδων. Στο πρώτο επίπεδο παρουσιάζονται τα ονόματα όλων των αρχείων των προγραμμάτων και στο δεύτερο επίπεδο ο αύξων αριθμός, η ημερομηνία και η ώρα της κάθε έκδοσης. Οι γραμμές του πηγαίου κώδικα είναι αριθμημένες για την ευκολότερη μελέτη των εκδόσεων ενός προγράμματος. Η δυνατότητα της καταγραφής των ενεργειών των μαθητών επιτρέπει: (1) την καταγραφή των δυσκολιών και των παρανοήσεων των μαθητών κατά την εισαγωγή τους στον αντικειμενοστρεφή προγραμματισμό, (2) τη διερεύνηση των στρατηγικών επίλυσης προβλημάτων των μαθητών, αφού κάθε έκδοση του προγράμματος αντιπροσωπεύει ένα βήμα της διαδικασίας επίλυσής τους, και (3) στο διδάσκοντα να προσαρμόσει το μάθημα στις ανάγκες των μαθητών.



Εικόνα 4. Παρουσίαση των εκδόσεων και των αντίστοιχων αποτελεσμάτων μεταγλώττισης ενός προγράμματος.



Δραστηριότητα 5

Συγκρίνετε τους μικρόκοσμους JKarelRobot, Jeroo και objectKarel:

- καταγράφοντας τα πλεονεκτήματα και τα μειονεκτήματά τους
- τον βαθμό στον οποίο πιστεύετε ότι κάθε ένας από τους τρεις μικρόκοσμους βοηθάει στην αντιμετώπιση των γενικών δυσκολιών που αναφέρονται στην αρχή της ενότητας 7
- αναφέροντας σε ποια περίπτωση θεωρείτε ότι είναι πιο κατάλληλος κάθε ένας από τους τρεις μικρόκοσμους.

Εκπαιδευτικά περιβάλλοντα προγραμματισμού

Όπως ήδη αναφέρθηκε, για τη διδασκαλία του αντικειμενοστρεφούς προγραμματισμού σε μαθητές προτείνεται η χρήση ενός μικρόκοσμου. Ωστόσο, στην περίπτωση που η διδασκαλία των αρχών του αντικειμενοστρεφούς προγραμματισμού γίνει σε μαθητές με εμπειρία στον προγραμματισμό, όπως για παράδειγμα σε μαθητές Τεχνικών/Επαγγελματικών Λυκείων ή Σχολών σχετικών ειδικοτήτων μπορεί να χρησιμοποιηθεί ένα εκπαιδευτικό προγραμματιστικό περιβάλλον που υποστηρίζει μια συμβατική γλώσσα προγραμματισμού. Το πιο δημοφιλές εκπαιδευτικό περιβάλλον προγραμματισμού είναι το BlueJ (<http://www.bluej.org>), το οποίο χρησιμοποιεί τη Java.

Ιδιαίτερο ενδιαφέρον παρουσιάζει επίσης το περιβάλλον Greenfoot (<http://www.greenfoot.org>), το οποίο βασίστηκε στο BlueJ χρησιμοποιεί ως γλώσσα προγραμματισμού τη Java, αλλά υιοθετεί στοιχεία των μικρόκοσμων προγραμματισμού. Συγκεκριμένα, δίνει τη δυνατότητα δημιουργίας σεναρίων που οπτικοποιούνται και μπορούν να χρησιμοποιηθούν ως πλαίσιο διδασκαλίας όλων των αντικειμενοστρεφών εννοιών.

8. ΕΝΔΕΙΚΤΙΚΕΣ ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ/ΣΕΝΑΡΙΑ

Στο αρχείο “Δραστηριότητες_BlueJ.pdf” παρουσιάζονται τρία φύλλα δραστηριοτήτων για την εισαγωγή των μαθητών στις βασικές έννοιες του αντικειμενοστρεφούς προγραμματισμού με το περιβάλλον BlueJ. Σε αυτά τα φύλλα δραστηριοτήτων παρουσιάζονται κάποιες ενδεικτικές εισαγωγικές δραστηριότητες για τη διδασκαλία των πιο βασικών αντικειμενοστρεφών εννοιών στο περιβάλλον του BlueJ, αξιοποιώντας τη διδακτική προσέγγιση που παρουσιάστηκε στις προηγούμενες ενότητες.

9. ΧΡΗΣΗ ΕΞΩΤΕΡΙΚΩΝ ΠΗΓΩΝ

Alphonse, C. & Ventura, P. (2002), Object-Orientation in CS1-CS2 by design, *Proceedings of the 7th Annual Conference on Innovation and Technology in Computer Science Education*, 70-74.

Bergin, J., Stehlik, M., Roberts, J. and Pattis, R. (1997), *Karel++ - A Gentle Introduction to the Art of Object-Oriented Programming*, John Wiley & Sons, New York

Bruce, K. Danyluk, A. & Murtagh, T. (2001), A Library to Support a Graphics-Based Objects-First Approach to CS1, *ACM SIGCSE Bulletin*, Vol. 33, No.1, 6-10.

Brusilovsky, P., Calabrese, E., Hvorecky, J., Kouchnirenko, A. and Miller P. (1997), Mini-languages: a way to learn programming principles, *Journal of Education and Information Technologies*, 2, 65-83

Buck, D. and Stucki, D.J. (2000), JKarelRobot: A Case Study in Supporting Levels of Cognitive Development in the Computer Science Curriculum, *ACM SIGCSE Bulletin*, 33(1), 16-20

Christensen, H. & Caspersen, M. (2002), Frameworks in CS1 - a Different Way of Introducing Event-driven Programming, *Proceedings of the 7th Annual Conference on Innovation and Technology in computer Science Education*, 75-79.

Decker, R. & Hirshfield, S. (1994), The Top 10 Reasons Why Object-Oriented Programming Can't Be Taught In CS1, *ACM SIGCSE Bulletin*, Vol. 26, No. 1, 51-55.

Fleury, A. E. (2001), Encapsulation and reuse as viewed by java students, *ACM SIGCSE Bulletin*, Vol. 33, Issue 1, 189-193.

Hadjerrouit, S. (1998), A Constructivist Framework for Integrating the Java Paradigm into the Undergraduate Curriculum, *ACM SIGCSE Bulletin*, Vol. 30, Issue 3, 105-107.

Hadjerrouit, S. (1999), A constructivist approach to object-oriented design and programming, *ACM SIGCSE Bulletin*, Vol. 31, Issue 3, 171-174.

Holland, S. Griffiths, R. & Woodman, M. (1997), Avoiding object misconceptions, *ACM SIGCSE Bulletin*, Vol. 29, No. 1, 131-134.

Kolling, M. & Rosenberg, J. (2001), Guidelines for Teaching Object Orientation with Java, *ACM SIGCSE Bulletin*, Vol. 33 Issue 3, 33-36.

Pattis, R. E., Roberts, J. and Stehlik, M. (1995), *Karel - The Robot, A Gentle Introduction to the Art of Programming* (2nd ed.), John Wiley & Sons, New York

Ragonis, N. & Ben-Ari, M. (2005), A Long-Term Investigation of the Comprehension of OOP Concepts by Novices, *International Journal of Computer Science Education*, 15(3), 203-221.

Sanders, D. and Dorn, B. (2003), Jeroo: A Tool for Introducing Object-Oriented Programming, *ACM SIGCSE Bulletin*, 35 (1), 201-204

Tempte, M C. (1991), Let's Begin Introducing the Object-Oriented Paradigm, *ACM SIGCSE Bulletin, Vol. 23, No. 1*, 338-342.

Wick, M. (1995), On Using C++ and Object-Orientation in CS1: the Message is still more important than the Medium, *ACM SIGCSE Bulletin, Vol. 27, Issue 1*, 322-326.

Xinogalos, S., and Satratzemi, M. (2005a), The Hands-on Activities of the Programming Microworld objectKarel, *In Proceedings of the 10th annual ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE 2005)*, 384

Xinogalos, S., and Satratzemi, M. (2005b), Using Hands-on Activities for Motivating Students with OOP Concepts Before They Are Asked to Implement Them, *In Proceedings of the 10th annual Conference on Innovation and Technology in Computer Science Education (ITiCSE 2005)*, 380

Xinogalos, S., Satratzemi, M. (2005) The Hands-on Activities of the Programming Microworld objectKarel, *ACM SIGCSE Bulletin, Vol. 37, Number 3*, September 2005, 384.

Xinogalos, S., Satratzemi, M. (2005) Using Hands-on Activities for Motivating Students with OOP Concepts Before They Are Asked to Implement Them, *ACM SIGCSE Bulletin, Vol. 37, Number 3*, September 2005, 380.

Xinogalos, S., Satratzemi, M. and Dagdilelis, V. An introduction to object-oriented programming with a didactic microworld: objectKarel, *Computers & Education, Volume 47, Issue 2*, September 2006, 148-171

Ξυνόγαλος, Σ. (2002), Εκπαιδευτική Τεχνολογία: Ένας Διδακτικός Μικρόκοσμος για την Εισαγωγή στον Αντικειμενοστραφή Προγραμματισμό, *Διδακτορική διατριβή*, Τμήμα Εφ. Πληροφορικής, Πανεπιστήμιο Μακεδονίας

Ξυνόγαλος, Σ. (2003) «Η διδακτική και μαθησιακή δραστηριότητα στον προγραμματιστικό μικρόκοσμο objectKarel», *Πρακτικά του 6ου Πανελληνίου Συνεδρίου με Διεθνή Συμμετοχή «Διδακτική των Μαθηματικών και Πληροφορική στην Εκπαίδευση»*, Βόλος, 10-13 Οκτωβρίου 2003, 317-326

Ξυνόγαλος, Σ. (2003) «Σενάρια Διδασκαλίας του Προγραμματισμού στη Δευτεροβάθμια Εκπαίδευση», *Πρακτικά του 2ου Πανελληνίου Συνεδρίου των Εκπαιδευτικών για τις ΤΠΕ «Αξιοποίηση των Τεχνολογιών της Πληροφορίας και της Επικοινωνίας στη Διδακτική Πράξη»*, Σύρος, 9-11 Μαΐου 2003, Α' τόμος, 783-795.

Ξυνόγαλος, Σ. (2006), Η διδασκαλία του αντικειμενοστρεφούς προγραμματισμού με τον μικρόκοσμο objectKarel: εμπειρίες, προβληματισμοί και προτάσεις, *Περιοδικό «ΘΕΜΑΤΑ*

στην Εκπαίδευση» ειδικό αφιέρωμα: «Σύγχρονη έρευνα στη Διδακτική της Πληροφορικής: ερευνητικοί άξονες, μέθοδοι, τεχνικές, εργαλεία», 7:3, 305-327, Ελληνικά γράμματα.

Ξυνόγαλος, Σ. (2008), Μελέτη των Δυσκολιών των Φοιτητών για την Έννοια του «Αντικειμένου» στον Αντικειμενοστραφή Προγραμματισμό, *Πρακτικά 4ου Πανελληνίου Συνεδρίου «Διδακτική της Πληροφορικής»*, Πάτρα 28-30 Μαρτίου, 91-100.

Ξυνόγαλος, Σ. (2012), Διδακτική Προσέγγιση για τον Αντικειμενοστρεφή Προγραμματισμό: εφαρμογή στο BlueJ, *Πρακτικά 6ου Πανελληνίου Συνεδρίου “Διδακτική της Πληροφορικής”*, Φλώρινα, 20-22 Απριλίου, 63-72.

Ξυνόγαλος, Σ. & Σατρατζέμη, Μ. (2004) Η Εισαγωγή στον Αντικειμενοστραφή Προγραμματισμό: Προβλήματα και Μεθοδολογίες για την Αντιμετώπισή τους, *Πρακτικά του 4ου Πανελληνίου Συνεδρίου με Διεθνή Συμμετοχή «Οι Τεχνολογίες της Πληροφορίας και της Επικοινωνίας στην Εκπαίδευση»*, Αθήνα, 29 Σεπτεμβρίου-3 Οκτωβρίου 2004, 133-142 (τόμος Β’).

Ξυνόγαλος, Σ., Σατρατζέμη, Μ. & Δαγδιλέλης, Β. (2000) Η εισαγωγή στον προγραμματισμό: Διδακτικές Προσεγγίσεις και Εκπαιδευτικά Εργαλεία, *Πρακτικά του 2ου Πανελληνίου Συνεδρίου με Διεθνή Συμμετοχή «Οι Τεχνολογίες της Πληροφορίας και της Επικοινωνίας στην Εκπαίδευση»*, Πάτρα, 13-15 Οκτωβρίου 2000, 115-124.

Σατρατζέμη, Μ., Ξυνόγαλος, Σ. & Δαγδιλέλης, Β. (2006), Εκπαιδευτικά Προγραμματιστικά Περιβάλλοντα για τη Διδασκαλία του Αντικειμενοστραφούς Προγραμματισμού: μια επισκόπηση, *Πρακτικά 5ου Πανελληνίου Συνεδρίου με Διεθνή Συμμετοχή «Οι Τεχνολογίες της Πληροφορίας και της Επικοινωνίας στην Εκπαίδευση»*, Θεσσαλονίκη, 5-8 Οκτωβρίου 2006, 899-906.

10. ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΠΕΡΑΙΤΕΡΩ ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ – ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΕΡΓΑΣΙΕΣ

- Αξιοποιώντας το υλικό που αναπτύχθηκε στα πλαίσια των προηγούμενων δραστηριοτήτων και εμπλουτίζοντάς το κατάλληλα, σχεδιάστε ένα ολοκληρωμένο φύλλο εργασίας με δραστηριότητες για την εισαγωγή στο αντικειμενοστρεφές παράδειγμα προγραμματισμού. Το φύλλο εργασίας μπορεί να αναφέρεται σε συγκεκριμένο λογισμικό, αλλά μπορεί επίσης να μην απαιτεί τη χρήση κάποιου

συγκεκριμένου εργαλείου.

- Αιτιολογήστε την επιλογή των δραστηριοτήτων που συμπεριλάβατε στο φύλλο εργασίας.
- Επιλέξτε κάποιο από τα εργαλεία που παρουσιάστηκαν, ή οποιοδήποτε άλλο της δικής σας επιλογής, και περιγράψτε με ποιο τρόπο θα αξιοποιούσατε τα ιδιαίτερα χαρακτηριστικά του για την αποτελεσματικότερη διδασκαλία των βασικών εννοιών του αντικειμενοστρεφούς προγραμματισμού. Περιγράψτε εισαγωγικές δραστηριότητες και τα αναμενόμενα οφέλη, λαμβάνοντας υπόψη τις δυσκολίες, παρανοήσεις και διδακτικά προβλήματα που έχουν καταγραφεί στη βιβλιογραφία.